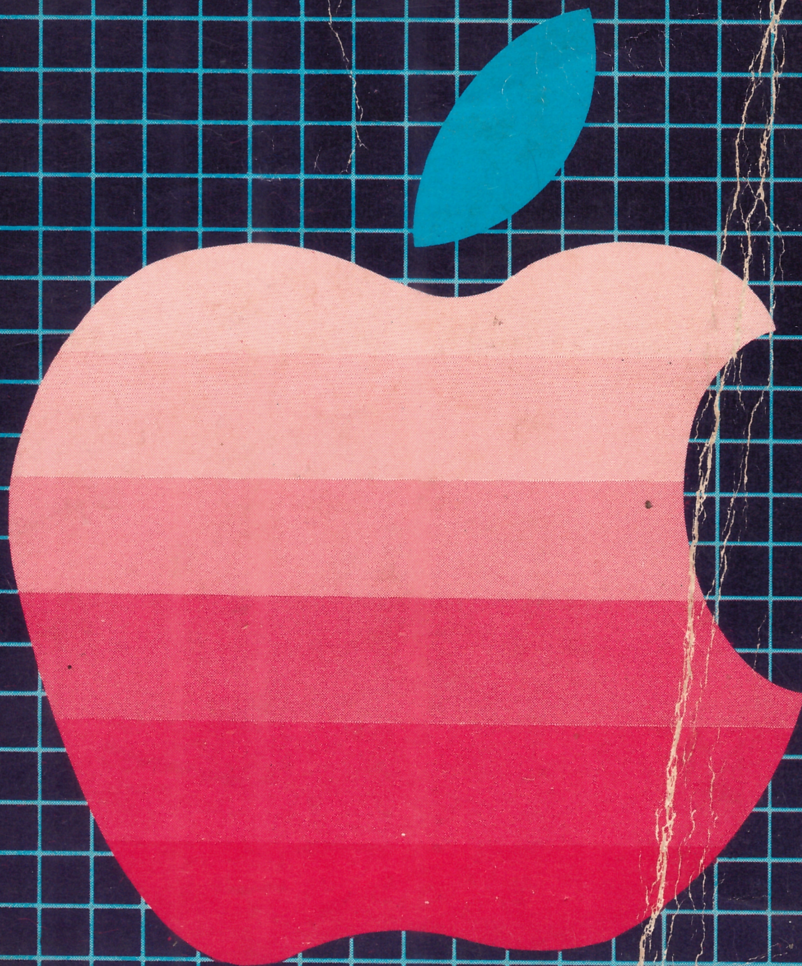
เรียน APPLESOFT BASIC ด้วยตนเอง

ประพัฒน์ อุทโยภาค



ซีเอ็ด

เรียน APPLESOFT BASIC ด้วยตนเอง

ประพัฒน์ อุทโยภาศ



บริษัท ซีเอ็ดยูเคชั่น จำกัด
SCIENCE, ENGINEERING & EDUCATION CO.,LTD.

ราคา 45 บาท

ISBN 974-7964-77-5

พิมพ์ครั้งที่ 1 พ.ศ. 2526

พิมพ์ครั้งที่ 2 พ.ศ. 2526

พิมพ์ครั้งที่ 3 พ.ศ. 2526

พิมพ์ครั้งที่ 4 พ.ศ. 2526

พิมพ์ครั้งที่ 5 พ.ศ. 2526

พิมพ์ครั้งที่ 6 พ.ศ. 2527

พิมพ์ครั้งที่ 7 พ.ศ. 2527

จัดพิมพ์และจัดจำหน่ายโดย



บริษัท ซีเอ็ดยูเคชั่น จำกัด
SCIENCE, ENGINEERING & EDUCATION CO.,LTD.

800/43-45 ซอยตระกูลสุข ถนนอโศกดินแดง
เขตห้วยขวาง กท. 10400 โทร.245-0390

พิมพ์ที่ หจก. เอช-เอน การพิมพ์
402/16 ซอยดอนกุศล เขตยานนาวา กทม. 10120
นายประพันธ์ จิระมงคล ผู้พิมพ์ผู้โฆษณา พ.ศ. 2527

คำนำ

สิ่งจูงใจที่ทำให้ผู้เขียนเขียนหนังสือเล่มนี้ขึ้นมาก็คือ ความอึดอัดใจ สมัยผู้เขียนเรียนในมหาวิทยาลัย คอมพิวเตอร์เป็นของวิเศษที่น้อยคนจะเข้าสัมผัสได้ ต่อมาถึงสมัยที่คอมพิวเตอร์เป็นของสามัญที่นักเรียนชั้นมัธยมในบางประเทศเรียนอยู่ในชั้นเรียน ผู้เขียนก็เห็นความจำเป็นที่จะต้องนำคอมพิวเตอร์มาใช้ในอาชีพ เมื่อภาระทางด้านอาชีพทำให้ไม่สามารถจะหาเวลาเข้าเรียนตามชั้นเรียนได้ ก็จำเป็นต้องเรียนด้วยตนเองโดยที่ไม่มีพื้นฐานความรู้ทางนี้อยู่เลยแม้แต่น้อย ผู้เขียนจึงประสบภาวะที่น่าอึดอัดใจดังกล่าว กล่าวคือ ไม่มีหนังสือใดที่จะสนองความประสงค์ได้เลย ไม่ยากเกินไปก็สับสนไม่เรียงตามลำดับ โปรแกรมตัวอย่างที่ให้ก็มีที่ผิดโดยเฉพาะคู่มือของผู้ผลิตเครื่องคอมพิวเตอร์

หนังสือเล่มนี้ มีวัตถุประสงค์ที่จะให้ความรู้แก่ผู้ที่ไม่มีพื้นฐานทางคอมพิวเตอร์มาก่อนให้สามารถจะเขียนโปรแกรมได้ สามารถใช้คอมพิวเตอร์เพื่อผลประโยชน์ในการทำงานและในการให้ความบันเทิง แต่มิได้มีจุดมุ่งหมายที่จะทำให้ผู้อ่านสามารถประกอบอาชีพในการเขียนโปรแกรม(Programmer)หรือมิได้มีจุดมุ่งหมายที่จะใช้เป็นเอกสารอ้างอิง สำหรับผู้ที่ได้เล่าเรียนมาทางด้านคอมพิวเตอร์โดยตรง ผู้ที่จะได้รับประโยชน์จากหนังสือเล่มนี้ คือ ผู้ที่อยากจะเรียนรู้เรื่องคอมพิวเตอร์แต่ไม่มีเวลาที่จะเข้าเรียน สามารถจะเรียนด้วยตนเองจนมาใช้ให้เกิดประโยชน์ได้

วัตถุประสงค์อีกสองประการที่ทำให้ผู้เขียนเขียนหนังสือเล่มนี้ขึ้นมา ก็คือ ประการแรก ผู้เขียนใคร่ที่จะส่งเสริมให้เยาวชนของชาติเป็นคนที่มี "คุณภาพ" ให้เท่าเทียมเยาวชนของชาติอื่นๆ ความรู้ทางเทคนิคสมัยใหม่ เช่น ความรู้ด้านคอม

พิวเตอร์ จะเป็นส่วนสำคัญให้การพัฒนาประเทศชาติให้เป็นไปอย่างรวดเร็ว และวัตถุประสงค์ประการสุดท้าย คือ เป็นการสนองกิเลส "ความสนุก" ของผู้เขียนอันมิอาจจะตีค่าเป็นเงินตราได้

ข้อจำกัดของหนังสือเล่มนี้ มีอยู่ 2 ประการคือผู้อ่านจะต้องมีพื้นฐานภาษาอังกฤษบ้างพอสมควร เพราะศัพท์บางคำ ผู้เขียนไม่ประสงค์ที่จะแปลเป็นภาษาไทยเพราะยากที่จะหาคำที่มีความหมายตรงกัน โดยเฉพาะภาษาที่จะติดต่อกับเครื่องได้คือ ภาษา BASIC จะต้องเป็นภาษาอังกฤษ อีกประการหนึ่ง ผู้อ่านควรจะต้องมีเครื่องคอมพิวเตอร์ เพื่อจะดำเนินการทดลองโปรแกรมได้

ภาษา BASIC ในหนังสือนี้ทั้งหมดสามารถนำมาใช้กับเครื่อง Apple II ได้ทันที และโปรแกรมทั้งหมดที่เขียนไว้ในหนังสือนี้เขียนด้วยเครื่อง Apple II ผ่าน Epson Printer โดยได้ทำการทดสอบจนเป็นที่แน่ใจแล้วว่าสามารถให้ผลได้จริงๆ

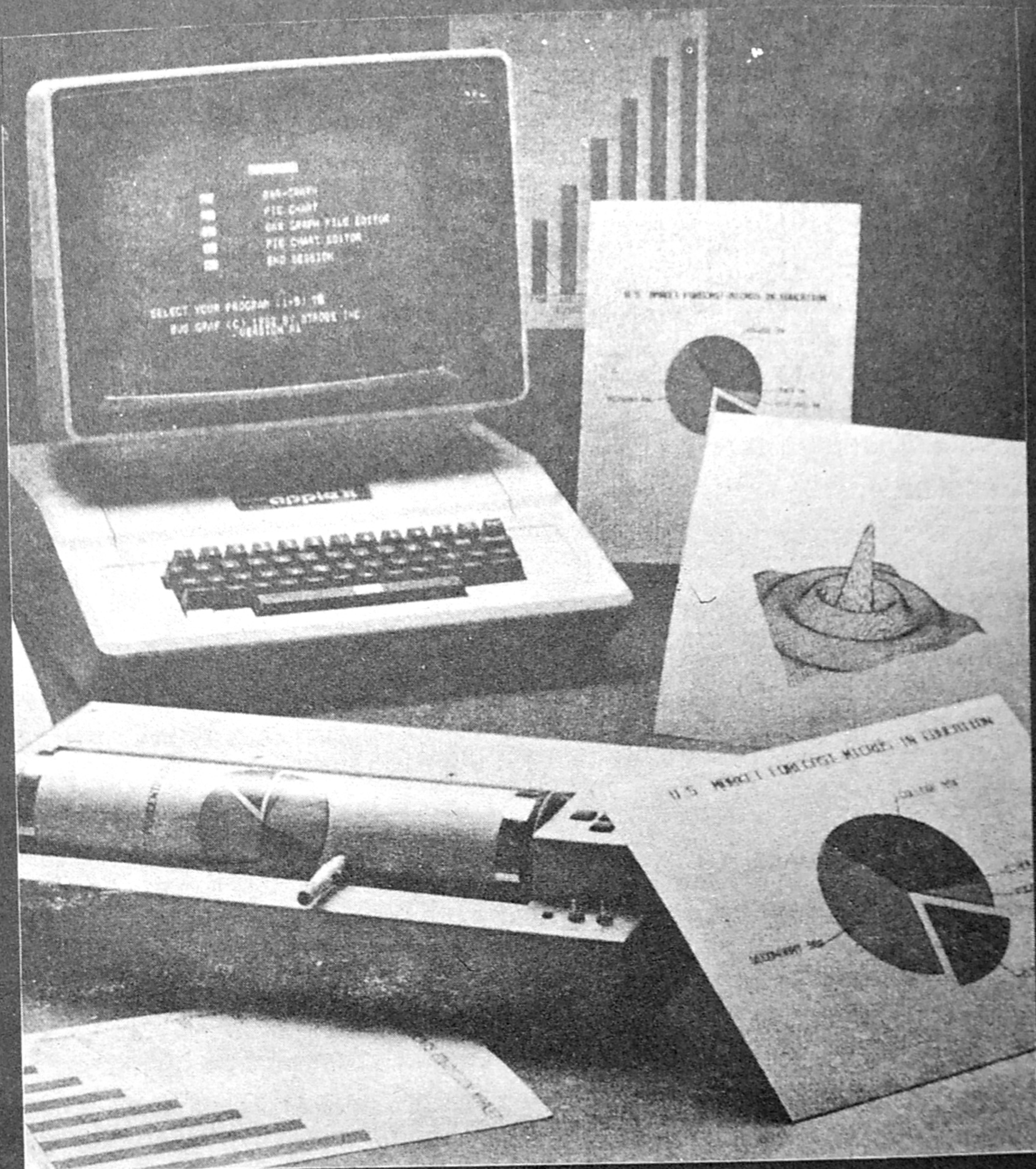
โปรแกรมเหล่านี้ ผู้เขียนเขียนขึ้นเองทั้งหมด แต่มีบางโปรแกรมอาจจะอาศัยเค้าโครงจากหนังสือบางเล่ม หากมีการผิดพลาดก็ขออภัย อนึ่ง ศัพท์ต่างๆที่ใช้บางคำผู้เขียนบัญญัติขึ้นเองตามความเหมาะสม(ในทัศนะของผู้เขียนเอง) ซึ่งจะเป็นที่ยอมรับหรือไม่ก็แล้วแต่วิจารณญาณของผู้ใช้

การที่หนังสือเล่มนี้สำเร็จขึ้นมาได้ ผู้เขียนเป็นหนี้บุญคุณบุคคลอยู่หลายคนซึ่งผู้เขียนต้องขอขอบคุณเป็นอย่างสูง โดยเฉพาะ น.พ. พันธ์พิษณุ์ สาครพันธ์ ผู้เป็นเพื่อนในวัยเด็ก เป็นครูผู้ให้ความรู้ด้านไมโครคอมพิวเตอร์ และเป็นผู้ตรวจทานหนังสือเล่มนี้ และแก้ไขให้ถูกต้องโดยตลอด คุณถนอม กลัดแก้ว กรรมการผู้จัดการ บริษัทกรุงเทพเอ็นยิเนียริ่งคอนซัลแตนท์ จำกัด พร้อมกับเพื่อนร่วมงานในบริษัทฯ นี้ที่สนับสนุน พร้อมกับให้อภัยในการที่ผู้เขียนเบียดบังเวลาทำงานเพื่อเขียนหนังสือเล่มนี้ และช่วยในการพิมพ์ต้นฉบับให้อย่างสวยงาม และสุดท้ายคือ ภุชงค์

บุตรชายคนเล็กของผู้เขียนที่ให้คำติชมบางตอนที่เป็นประโยชน์ทั้งๆที่ตอนเมื่อผู้เขียนเปรยๆ ว่าจะเขียนหนังสือเล่มนี้ ท้วงว่า "ไหวเหรอพ่อ" และเมื่อได้อ่านต้นฉบับตลอดแล้ว ก็ได้แต่พูดว่า "ก็ดีนี่พ่อ" แต่ในทางตรงข้ามหากบุตรชายคนนี้ไม่แย่งผู้เขียนใช้เครื่องไมโครคอมพิวเตอร์ หนังสือเล่มนี้คงจะสำเร็จได้เร็วกว่านี้

ในอนาคตที่เครื่องไมโครคอมพิวเตอร์จะเป็นอุปกรณ์การเรียนในชั้นมัธยมในประเทศไทยเรา ผู้เขียนหวังว่าหนังสือเล่มนี้คงจะเป็นประโยชน์แก่นักเรียนไม่มากก็น้อย

ประพัฒน์ อุทโยภาศ


PIE CHART
PIE CHART EDITOR
END SESSION
SELECT YOUR PROGRAM

สารบัญ

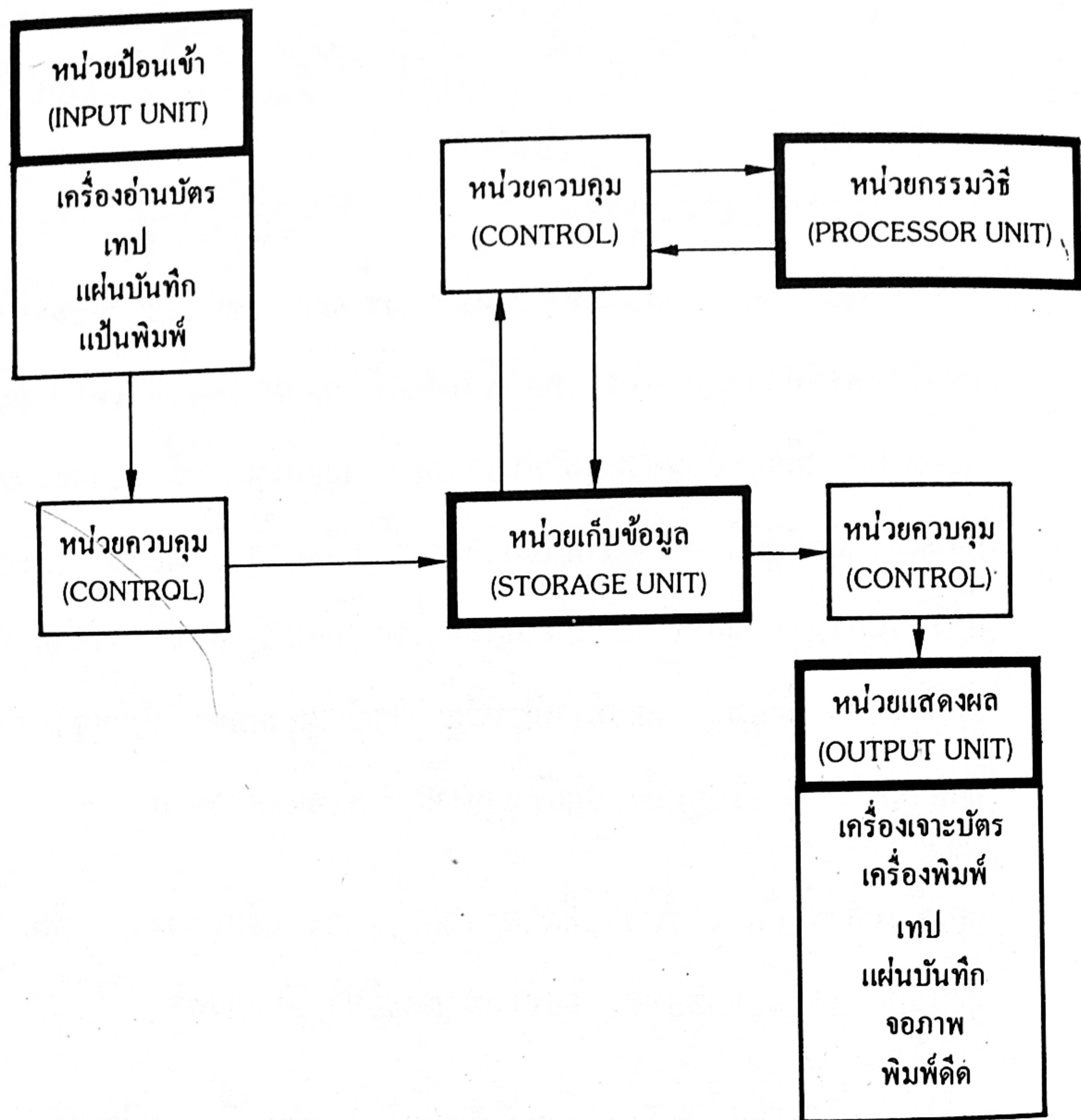
1 คอมพิวเตอร์

1.1 **คอมพิวเตอร์** คือ เครื่องมือที่มนุษย์สร้างขึ้นมา แต่ถึงแม้ว่าจะสร้างขึ้นมาอย่างวิจิตรพิศดารอย่างไร คอมพิวเตอร์ก็ไม่สามารถจะคิดได้เหมือนมนุษย์ มนุษย์เท่านั้นที่จะเป็นผู้ควบคุมคอมพิวเตอร์ มนุษย์เท่านั้นที่จะวางแนวทางให้คอมพิวเตอร์ปฏิบัติ แนวทางดังกล่าวนี้ เรียกว่า *โปรแกรม* (Program) ถ้าโปรแกรมใดไม่สมเหตุสมผล คอมพิวเตอร์ก็จะปฏิเสธที่จะทำงาน เปรียบเทียบง่ายๆ มีคอมพิวเตอร์ก็เหมือนมีผู้รับใช้ที่ปัญญาอ่อน แต่มีความจำดีเลิศ หากรู้จักใช้ ผู้รับใช้ผู้นี้ก็จะปฏิบัติงานให้อย่างได้ผลมหาศาล

คอมพิวเตอร์ คือ เครื่องมือที่สามารถแก้ปัญหา, เขียนภาพ, พิมพ์อักษร, เก็บรักษาและค้นหาข้อมูล, เล่นเกม และอื่นๆ อีกสารพัด

งานบางอย่างเป็นการง่ายมากสำหรับคนจะทำ แต่เครื่องไม่มีความสามารถจะทำได้ แต่งานบางอย่างที่ถ้าหากใช้คนทำ อาจจะต้องใช้เวลานาน และมีการผิดพลาด แต่คอมพิวเตอร์จะทำได้ในเวลาอันรวดเร็ว และถูกต้องทุกประการ แต่ถึงอย่างไรก็ตาม คอมพิวเตอร์จะไม่ฉลาดเกินไปกว่าคนวางโปรแกรมอย่างแน่นอน

1.2 **ระบบคอมพิวเตอร์** รูปที่ 1.1 คือ แผนผังอย่างง่ายๆ ของระบบคอมพิวเตอร์ ซึ่งไม่ว่าคอมพิวเตอร์ขนาดใหญ่ ราคาเป็นล้านๆ บาท หรือไมโครคอมพิวเตอร์ ราคาไม่กี่พันบาท จะมีระบบเหมือนกัน ดังแสดงในรูปที่ 1.1



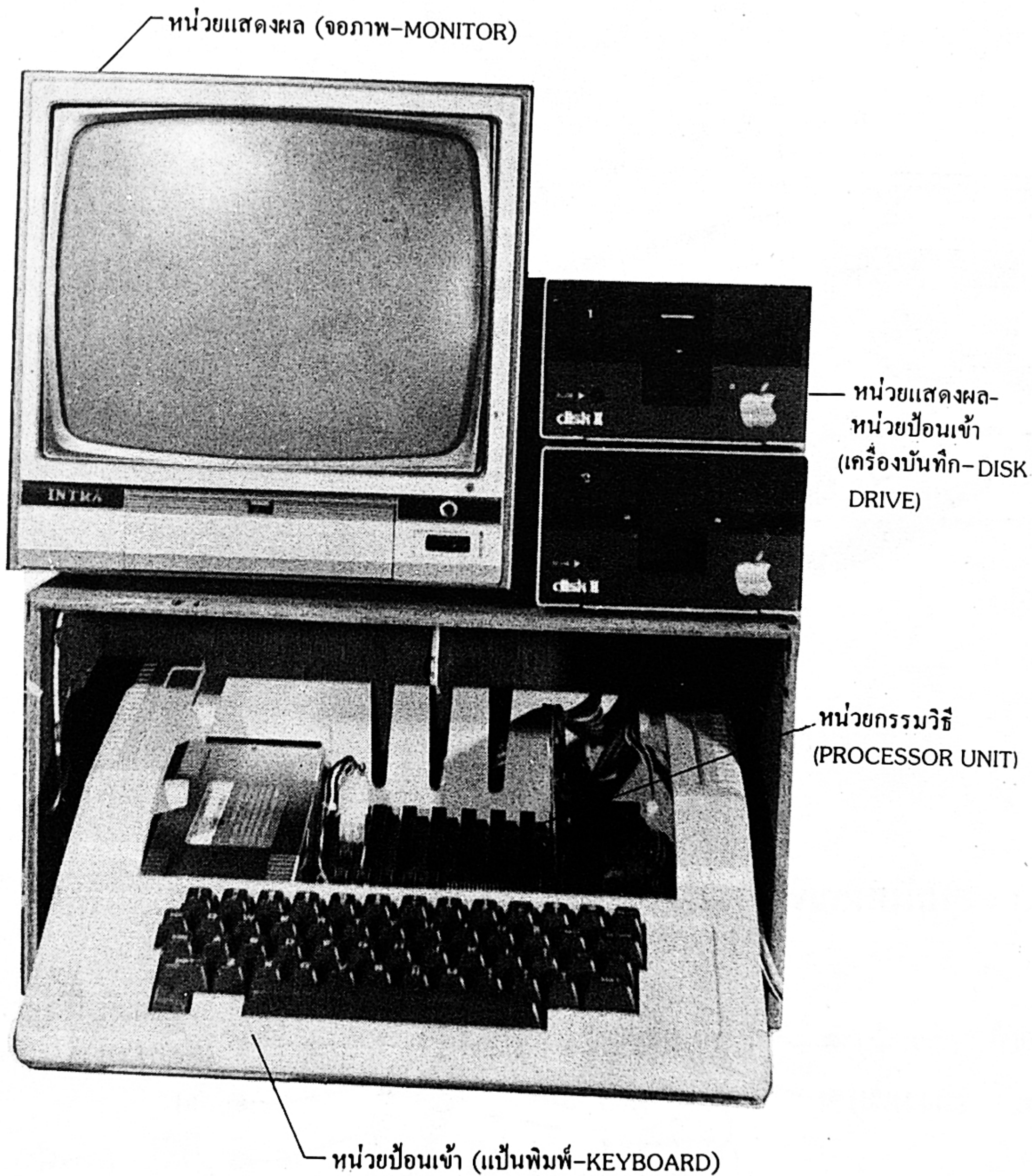
รูปที่ 1.1 ระบบคอมพิวเตอร์

หน่วยป้อนเข้า (INPUT UNIT) เป็นทางที่มนุษย์จะติดต่อกับคอมพิวเตอร์ โดยการป้อนข้อมูลเข้า หรือให้คำสั่งให้คอมพิวเตอร์ปฏิบัติ

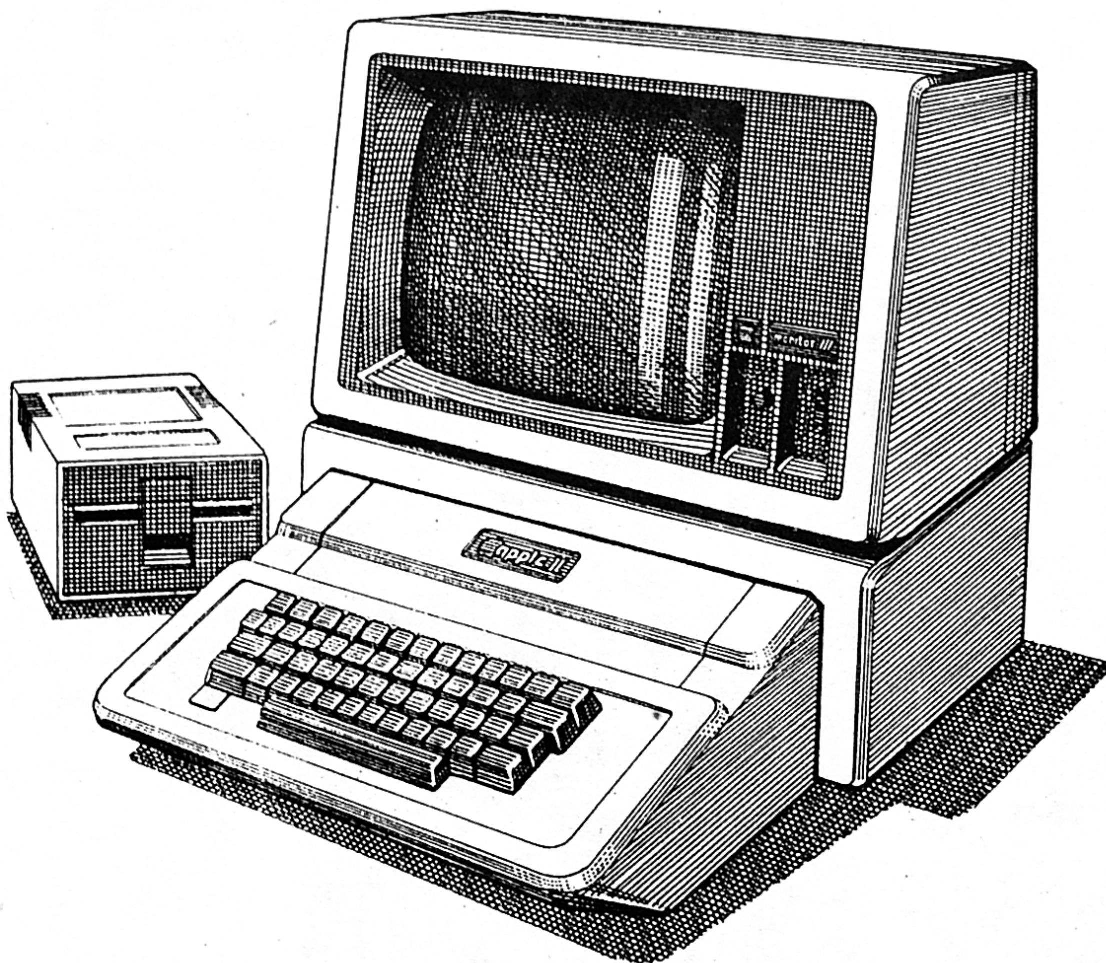
หน่วยเก็บข้อมูล (STORAGE UNIT) จะทำหน้าที่เก็บข้อมูล ข่าวสาร และคำสั่งไว้จนกว่าจะนำมาใช้เมื่อต้องการ

หน่วยกรรมวิธี (PROCESSOR UNIT) จะแปลคำสั่ง และควบคุมการปฏิบัติงาน และจะดำเนินการคำนวณ

หน่วยแสดงผล (OUTPUT UNIT) จะให้ผลลัพธ์และคำตอบ ซึ่งอาจจะทางจอภาพ (VIDEO) หรือเครื่องพิมพ์ (PRINTER)

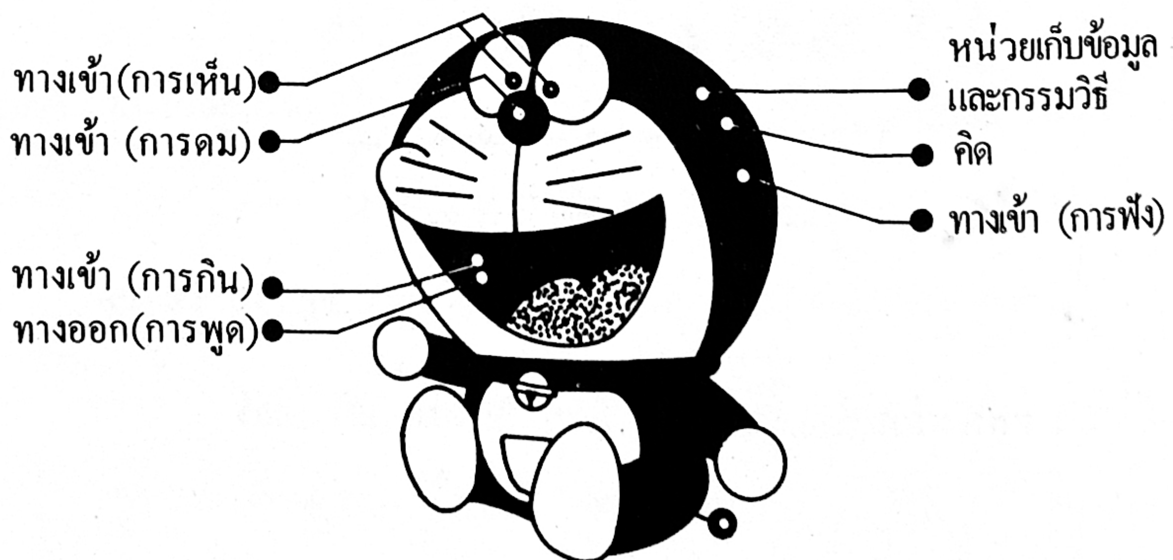


รูปที่ 1.2 แสดงส่วนต่าง ๆ และหน้าที่ของไมโครคอมพิวเตอร์



รูปที่ 1.3 ไมโครคอมพิวเตอร์ APPLE II

1.3 คนกับคอมพิวเตอร์



รูปที่ 1.4 คนเปรียบเทียบกับคอมพิวเตอร์

มนุษย์ต่างกับคอมพิวเตอร์ตรงที่มนุษย์คิดเองได้ และที่สำคัญมนุษย์เป็นผู้สร้างคอมพิวเตอร์ อย่ายกย่องหรือเกิดความกลัวว่า คอมพิวเตอร์จะเหนือมนุษย์ แต่คอมพิวเตอร์เปรียบเสมือนมนุษย์ต่างดาวที่พูดกันคนละภาษากับคน คนพูดภาษาคน แต่คอมพิวเตอร์พูดภาษาเครื่องจึงมีความจำเป็นที่จะต้องหาภาษาที่พูดกันรู้เรื่อง มีผู้คิดภาษานี้ขึ้นมาหลายภาษา ซึ่งแต่ละภาษาก็มีความสามารถที่จะติดต่อกันได้แตกต่างกันไป BASIC เป็นอีกภาษาหนึ่งที่ใช้พูดกับเครื่องโดยผ่านการแปลซึ่งปัจจุบันภาษา BASIC เป็นภาษาที่ใช้กันอย่างแพร่หลาย และในหนังสือนี้ จะกล่าวถึงภาษานี้ภาษาเดียว



ภาษา BASIC ย่อมาจาก

B = Beginner's

A = All Purpose

S = Symbolic

I = Instruction

C = Code

1.4 การเปรียบเทียบภาษามนุษย์กับภาษา BASIC

คอมพิวเตอร์จะรู้จักคำในภาษา BASIC อยู่ประมาณ 100 คำ มนุษย์ที่จะต้องติดต่อกับคอมพิวเตอร์เพื่อใช้งาน จึงจำเป็นที่จะต้องรู้จักคำเหล่านี้ โดยเฉพาะคำที่เรียกว่าคำสั่ง (COMMANDS)

ภาษามนุษย์		ภาษา BASIC	
คำ	– ใช้เพื่อผูกประโยค	KEYWORDS	– ใช้เพื่อทำ STATEMENTS
ประโยค	– ใช้เพื่อเขียนข้อความ	STATEMENTS	– ใช้เพื่อสร้าง PROGRAM
ข้อความ	– ความยาวไม่จำกัด	PROGRAM	– ความยาวเปลี่ยนแปลงได้ แต่จำกัดที่ความสามารถของเครื่อง
คำสั่ง	– สั่งให้ผู้ปฏิบัติทำตาม	COMMANDS	– สั่งให้เครื่องทำงาน เช่น NEW, LIST, RUN
หมายเลขของประโยค	– มีก็ได้ ไม่มีก็ได้	LINE NUMBERS	– จำเป็นต้องมีทุก STATEMENTS

รูปที่ 1.5 ภาษามนุษย์กับภาษา BASIC

1.5 COMMANDS มี 2 แบบคือ SYSTEM COMMANDS คือคำสั่งที่คนสั่งให้เครื่องทำโดยตรง เช่น

NEW - ให้ลบความจำทั้งหมดในเครื่อง

LIST - ให้แสดงโปรแกรมที่เขียนไว้ในเครื่อง

RUN - ให้เครื่องปฏิบัติ (EXECUTE) ตามคำสั่ง

PRINT - ให้เครื่องแสดงผลออกมา

ฯลฯ

อีกแบบหนึ่ง คือ PROGRAM COMMAND คือ คำสั่งที่บรรจุไว้ในโปรแกรมแล้วให้เครื่องปฏิบัติตามโปรแกรมที่วางไว้ตามลำดับ เมื่อสั่งให้เริ่มทำงานตามโปรแกรม เช่น

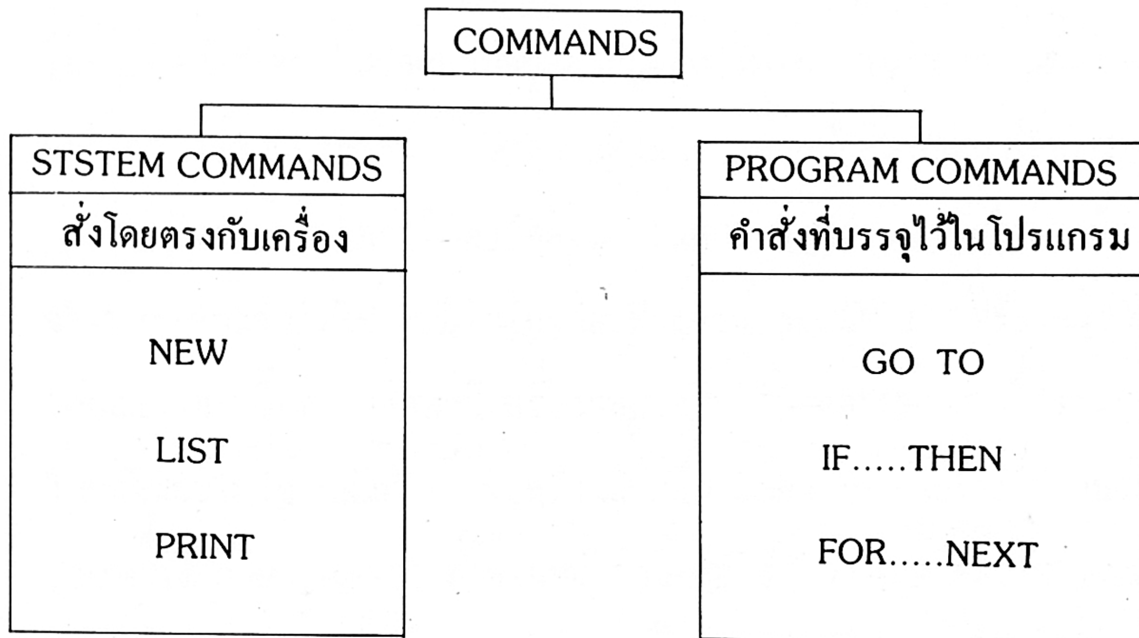
GO TO ...

IF ... THEN

FOR ... NEXT

INPUT

คำสั่งต่างๆ เหล่านี้ เราจะต้องเรียนรู้เป็นลำดับไป



รูปที่ 1.6 COMMANDS

1.6 **IMMEDIATE EXECUTION COMMAND และ DEFERRED EXECUTION COMMAND** เราอาจจะเรียก COMMAND เป็นอีกแบบหนึ่งได้ตามการสนองตอบของเครื่อง คือหากเราให้ COMMAND นั้นแก่เครื่องผ่านแป้นพิมพ์ หรือทางเข้าทางอื่นแล้ว เครื่องจะทำงานให้ทันที เราเรียก COMMAND นี้ว่า IMMEDIATE EXECUTION COMMANDS ส่วน COMMAND ที่ทำให้เครื่องทำงาน ก็ต่อเมื่อเครื่องถูกสั่งให้ดำเนินการตามโปรแกรมที่วางไว้ เราเรียกว่า DEFERRED EXECUTION COMMAND COMMAND บางคำเป็นได้ทั้ง IMMEDIATE EXECUTION COMMAND และ DEFERRED EXECUTION COMMAND เช่น PRINT, HOME แต่ COMMAND บางคำเป็น

ได้แต่เพียงอย่างใดอย่างหนึ่ง เช่น RUN เป็น IMMEDIATE EXECUTION COMMAND อย่างเดียว หรือ INPUT เป็นได้แต่ DEFERRED EXECUTION COMMAND อย่างเดียว

2 APPLE II MICROCOMPUTER

2.1 ไมโครคอมพิวเตอร์ในปัจจุบันมีอยู่หลายผู้ผลิต ไมโครคอมพิวเตอร์แต่ละแบบจะมีโครงสร้างและการทำงานคล้ายคลึงกัน ต่างกันที่รายละเอียดเฉพาะเครื่อง ซึ่งจะต้องทำการศึกษาโดยเฉพาะแต่ละแบบไป ในหนังสือเล่มนี้ จะจำเพาะเจาะจงให้ใช้ได้กับไมโครคอมพิวเตอร์ APPLE II หรือเครื่องที่เลียนแบบมา ซึ่งหากผู้ศึกษามีเครื่องแบบนี้จะนำโปรแกรมต่างๆ ที่ปรากฏอยู่ในหนังสือนี้ไปใช้ได้ทันที ผู้ที่มีเครื่องแบบอื่นจะต้องเปลี่ยนแปลงในรายละเอียดบ้าง ภาษา BASIC ที่ใช้ในหนังสือนี้มีชื่อเรียกว่า APPLESOFT BASIC .

2.2 **KEYBOARD ของ APPLE II** Keyboard หรือแป้นพิมพ์ คือ ประตูทางเข้าที่มนุษย์จะติดต่อกับคอมพิวเตอร์ โดยการป้อนข่าวสาร, ข้อมูล, ให้คำสั่งแก่คอมพิวเตอร์ แป้นพิมพ์ของคอมพิวเตอร์ เมื่อดูผาดๆ ก็เหมือนกับแป้นพิมพ์



รูปที่ 2.1 APPLE II KEYBOARD

คีดของพิมพ์ดีดธรรมดา ซึ่งก็เป็นจริงที่ตำแหน่งตัวอักษรต่างๆ บนแป้นพิมพ์ดีดของคอมพิวเตอร์ **อยู่ในตำแหน่ง**เดียวกับแป้นพิมพ์ดีดธรรมดามีบางแป้นที่แตกต่างออกไป และใช้งานแปลกไปกว่าพิมพ์ดีดธรรมดา ซึ่งจะได้เรียนรู้ในรายละเอียดต่อไป

แป้นตัวอักษรและตัวเลข (ALPHANUMERIC KEY) – แป้นตัวอักษร A ถึง Z และตัวเลข 1 ถึง 0 จะมีข้อแตกต่างไปจากแป้นพิมพ์ดีดธรรมดาเล็กน้อยคือ

แป้นพิมพ์ดีดธรรมดา	แป้นพิมพ์ดีดเครื่อง APPLE II
1. พิมพ์ดีดธรรมดาจะได้ตัวอักษรตัวเล็ก เมื่อยกแคร่ (กด SHIFT) ค้างไว้พร้อมกับพิมพ์จะได้ตัวอักษรตัวใหญ่	1. เมื่อกดตัวอักษร จะกด SHIFT หรือไม่ก็ตามจะได้ตัวอักษรตัวใหญ่ (CAPITAL) เสมอ นอกจากอักษร P,N และ M จะได้ตัว @, ^ และ]
2. สำหรับแป้นตัวเลขและเครื่องหมายอื่นที่มีรูปแสดงไว้ข้างบนและข้างล่าง เมื่อพิมพ์ดีดธรรมดาจะได้ตัวล่าง เมื่อกดยกแคร่จะได้ตัวบน	2. เหมือนพิมพ์ดีดธรรมดา
3. เลข 1 อาจจะใช้ตัว l เล็กแทน และ 0 (ศูนย์) กับ O (โอ) จะคล้ายกัน	3. เลข 1 กับ l แยกแป้นกัน เลข 0 (ศูนย์) จะทำให้ดูต่างจาก O (โอ) ด้วยการขีดเครื่องหมายทะแยงคล่อม ดังนี้ ø
4. เครื่องหมายคูณจะใช้ตัว x	4. เครื่องหมายคูณใช้ *
5. เครื่องหมายหารใช้ ÷	5. เครื่องหมายหารใช้ /

รูปที่ 2.2 แป้นพิมพ์ดีดธรรมดากับแป้นพิมพ์ดีดเครื่อง APPLE II

เครื่องหมายอื่นที่แตกต่างไปจากแป้นพิมพ์ดีดธรรมดา

CTRL	เรียกว่า CONTROL KEY ต้องใช้ร่วมกับแป้นอื่น โดยการกดแป้นนี้ค้างไว้ก่อน แล้วจึงกดแป้นอื่น
ESC	เป็นคำย่อของ ESCAPE การใช้ต่างกับ CONTROL KEY โดยการกดแป้นนี้ปล่อยออก แล้วกดแป้นอื่นต่อไป
REPT	ย่อมาจาก REPEAT ใช้สำหรับเวลาต้องการพิมพ์คำอื่นซ้ำๆ โดยการกดแป้นคำนั้นก่อน แล้วจึงกด REPT ปล่อยมือ เมื่อต้องการจะหยุด
RESET	สำหรับ APPLE II ที่มี Autostart ROM การกด RESET เป็นการย้อนไปหาต้นของ Program
RETURN	เป็นแป้นที่ใช้บ่อยที่สุด เมื่อกดแป้นนี้จะเป็นการบอกเครื่องว่าได้จบบรรทัดที่ได้พิมพ์ไปแล้ว และจะเริ่มบรรทัดใหม่ หากมี COMMAND ใดก่อนกด RETURN ก็ให้เครื่องปฏิบัติตาม COMMAND ได้
SHIFT	เหมือนกับแป้นยกแคร่ในพิมพ์ดีด เพื่อใช้พิมพ์เครื่องหมายที่อยู่ส่วนบนของแป้น ดังที่กล่าวไว้ใน "แป้นตัวอักษรและตัวเลข"
←	เป็นการถอยหลังกลับซึ่งจะลบอักษร หรือเครื่องหมายที่ได้พิมพ์ไปแล้วออกจากหน่วยความจำ
→	คล้ายๆ กับคานเคาะ(SPACE BAR) ต่างกันแต่ว่าเมื่อเลื่อนไปตรงกับอักษรหรือเครื่องหมายใดบนจอภาพ เท่ากับได้พิมพ์

อักษร หรือเครื่องหมายนั้นลงไป เครื่องจะรับอักษร หรือเครื่องหมายนั้นเข้าหน่วยความจำ

| $ | แทนคำว่า STRING ดูบทที่ 15

| < , | เครื่องหมายบน(พิมพ์เมื่อยกแคร่) เป็นเครื่องหมายที่มีความหมายว่า "น้อยกว่า"

| > . | เครื่องหมายบน เป็นเครื่องหมายที่มีความหมายว่า "มากกว่า"

| ^ N | เครื่องหมายบน เป็นเครื่องหมายที่มีความหมายว่า "ยกกำลัง"

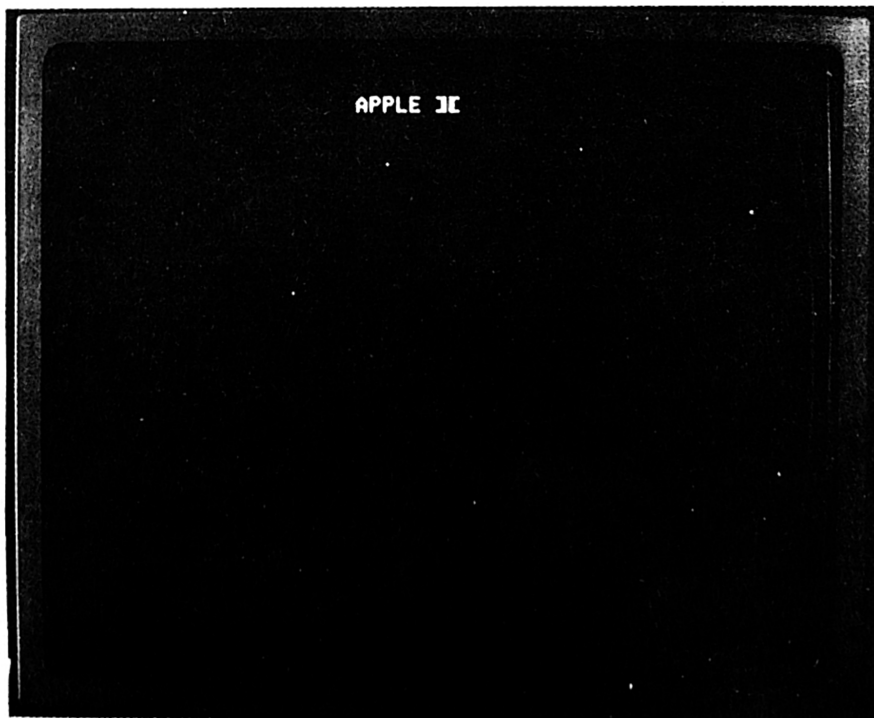
| = | เครื่องหมายเท่ากับ ทำหน้าที่ 2 หน้าที่ คือ "เท่ากับ" และทำหน้าที่บรรจุจำนวนเลขลงในตัวแปร เช่น พิมพ์ A = 2 หมายความว่า บรรจุจำนวน 2 ลงในตัวแปรที่มีชื่อว่า "A"

SPACE BAR หรือแป้นเว้นช่องไฟ คือ แป้นยาวที่อยู่ส่วนล่างสุดทำหน้าที่เว้นช่องไฟ

3 ไมโครคอมพิวเตอร์ เครื่องคิดเลขธรรมดา

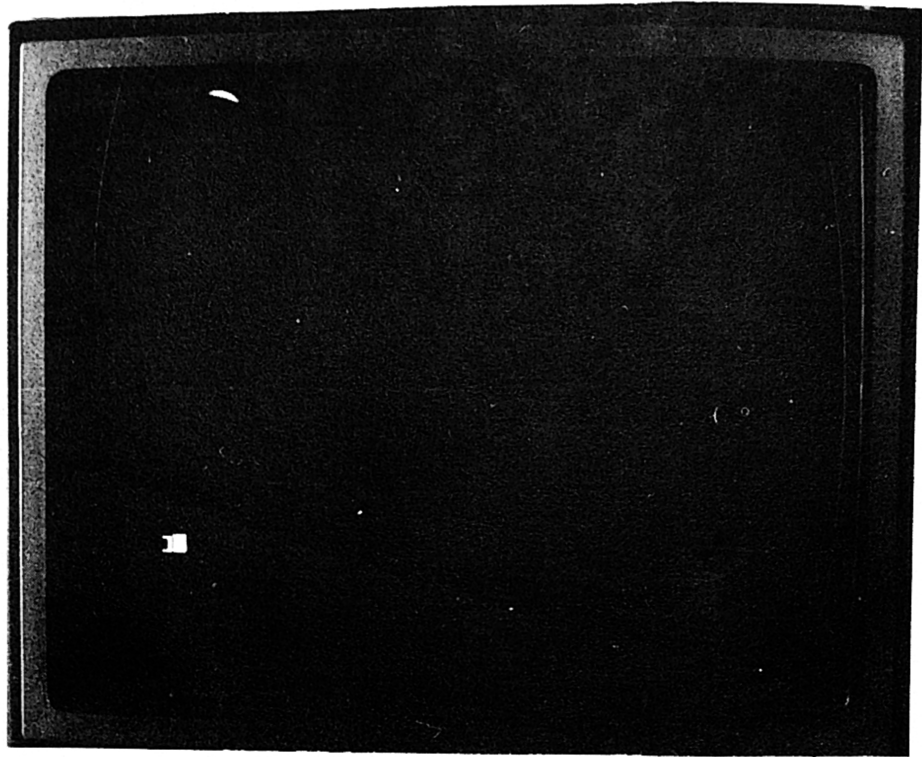
3.1 ในขั้นแรกนี้ เราจะลองใช้เครื่องไมโครคอมพิวเตอร์เป็นแบบเครื่องคิดเลขธรรมดาก่อน อุปกรณ์ที่ใช้ในขั้นนี้ คือ เครื่องคอมพิวเตอร์(ประกอบไปด้วย KEYBOARD – แป้นพิมพ์ ซึ่งติดตั้งรวมกับหน่วยกรรมวิธี และหน่วยเก็บข้อมูล) และจอภาพ

3.2 เมื่อตรวจสอบว่าการต่อเครื่องทุกอย่างเรียบร้อยแล้ว ก็เริ่มต้นด้วยการเปิดสวิทช์จอภาพ แล้วเปิดสวิทช์เครื่องซึ่งอยู่ด้านหลังของเครื่อง อักษร APPLE II จะปรากฏขึ้นที่ส่วนบนตอนกลางของจอภาพ พร้อมกับมีเสียงสั้นๆ ดังขึ้น ทางด้านบนซ้ายจะมีเครื่องหมายปรวกฏขึ้น



รูปที่ 3.1

หากมี DISK DRIVE ติดอยู่กับเครื่อง ตัว DISK DRIVE จะหมุนและไม่หยุด เมื่อเกิดกรณีเช่นนี้ ให้กด CONTROL RESET ตัว DRIVE จะหยุดลง พร้อมกับตัว APPLE II จะหายไป มีเครื่องหมายอีก 2 ชนิดเกิดขึ้นทางมุมล่างซ้ายของเครื่อง (ดูรูปที่ 3.2)



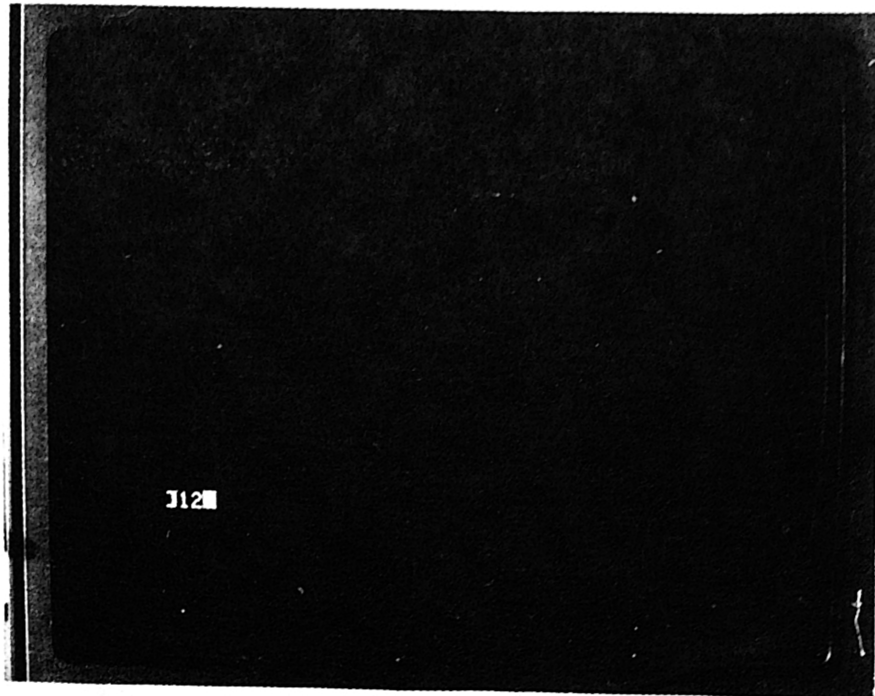
รูปที่ 3.2

เครื่องหมาย] เรียกว่า PROMPT CHARACTER ถ้ามีเครื่องหมายนี้ แสดงว่าเครื่องกำลังใช้ภาษา APPLESOFT BASIC อยู่

เครื่องหมาย █ สี่เหลี่ยมกระพริบ เรียกว่า CURSOR เป็นเครื่องหมายที่จะชี้บนจอภาพว่า ถ้ากดแป้นใดลงไป อักษรหรือเครื่องหมายนั้น จะปรากฏแทนที่ภาพสี่เหลี่ยมนั้น พร้อมกับตัวเครื่องหมาย CURSOR เอง จะขยับไปทางขวา 1 ตัวอักษร

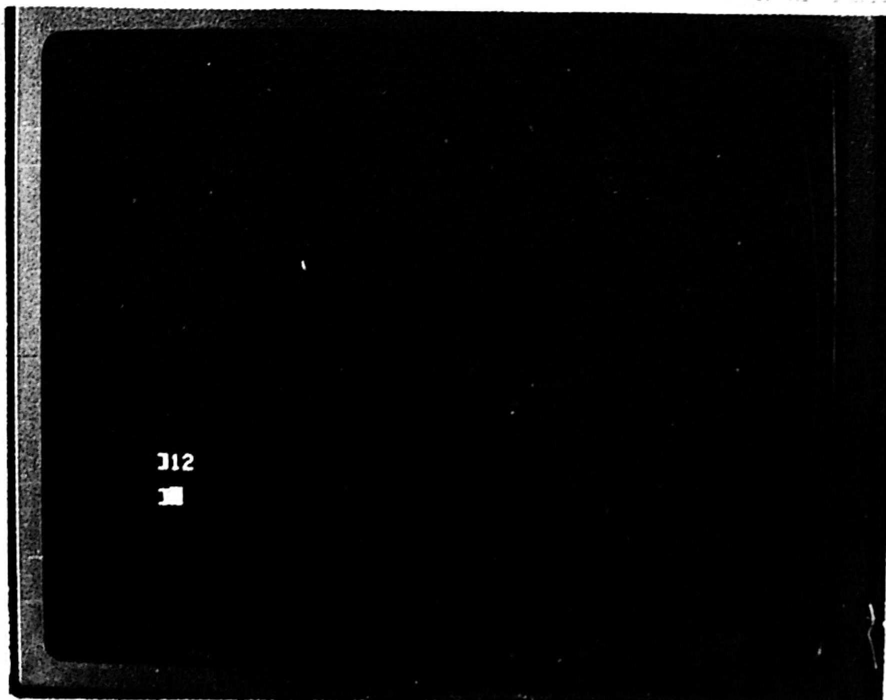
ลองกดแป้นเลข 1 และ 2 ตามลำดับ พร้อมกับสังเกตจอภาพ ทิ้งไว้สักพัก ดูว่าเครื่องจะทำอย่างไรต่อไป เครื่องจะไม่ทำอะไรทั้งนั้น เพราะ

เครื่องจะคอยว่าจะให้คำสั่งอย่างไรต่อไป เราจะกดตัวเลขเพิ่มขึ้นอีกหรือไม่ หรือจะจบสำหรับบรรทัดนั้น



รูปที่ 3.3

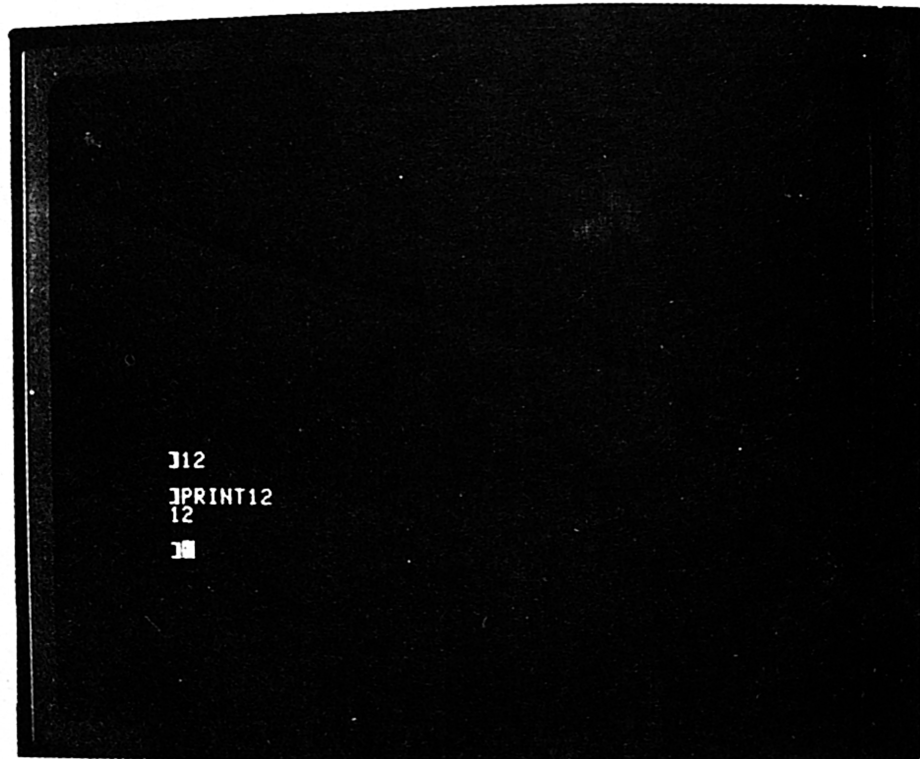
หากเราจะแจ้งแก่เครื่องว่า สำหรับบรรทัดนี้หมดแล้ว ให้กดแป้น RETURN ซึ่งเครื่องจะรับทราบแล้วย้อนกลับไปดูตรงต้นบรรทัดแล้วทำงานตามคำสั่ง



รูปที่ 3.4

เมื่อกดแป้น RETURN เครื่องหมาย PROMPT กับเลข 12 จะเลื่อนขึ้นไปบรรทัดบน ทิ้งเครื่องหมาย PROMPT พร้อมกับ CURSOR ไว้บนบรรทัดล่าง เนื่องจากยังไม่มีคำสั่ง เครื่องจึงไม่ให้คำตอบใดๆ

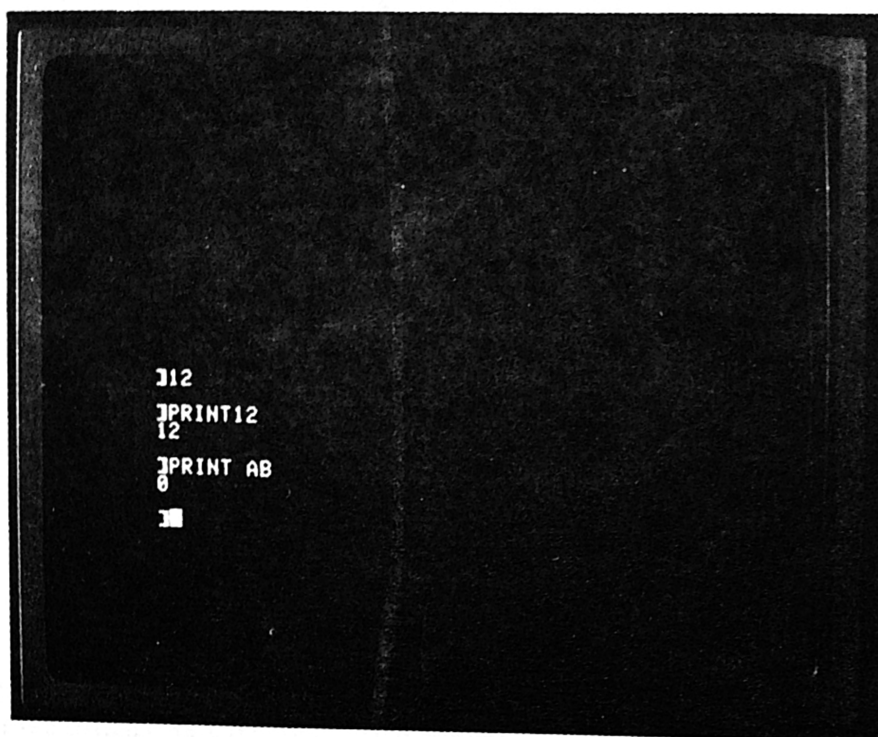
ต่อไปกด PRINT 12 ตอนนี้ เราสั่งเครื่องให้พิมพ์เลข 12 ลงไป เมื่อกด RETURN เครื่องจะพิมพ์เลข 12 ให้ในบรรทัดถัดไป สังเกตดูว่าหน้าคำตอบ จะไม่มีเครื่องหมาย PROMPT อยู่ บรรทัดที่พิมพ์แรกๆ จะขยับขึ้นข้างบน ส่วนเครื่องหมาย PROMPT และ CURSOR จะคงอยู่บรรทัดล่างสุด

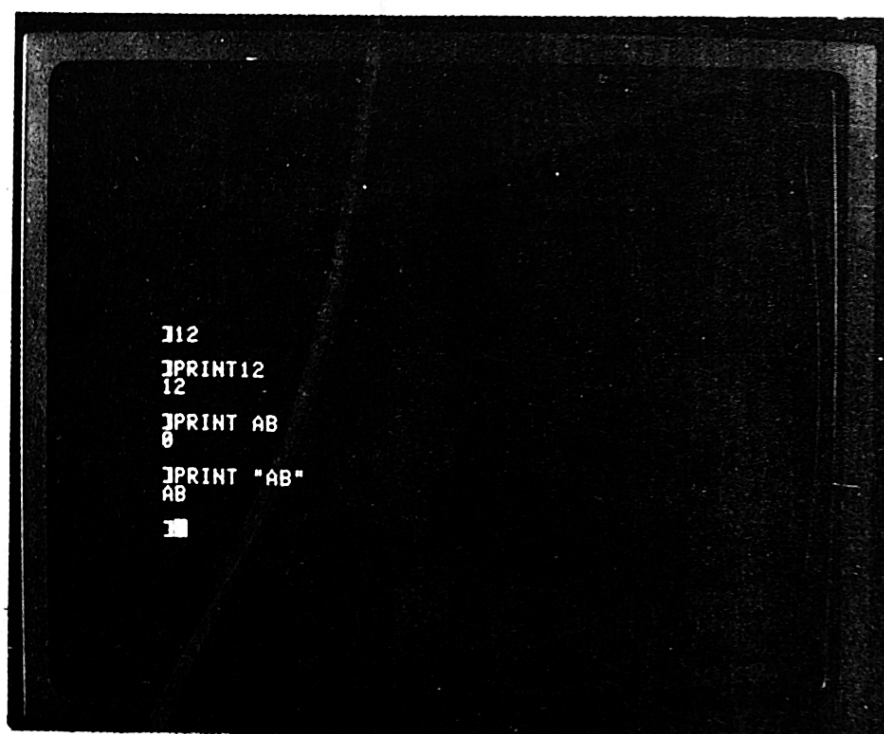


รูปที่ 3.5

ลองกด PRINT AB RETURN บรรทัดถัดมาจะเป็น 0 พร้อมกับ PROMPT และ CURSOR คงอยู่บรรทัดล่างสุด

ลองใส่เครื่องหมายคำพูดครอบ AB ลง PRINT "AB" RETURN เครื่องจะให้ AB ในบรรทัดถัดไป





รูปที่ 3.6

3.3 เครื่องหมายที่ใช้ในการคำนวณทั่วไปมีดังนี้

+	บวก
–	ลบ

×　　คูณ

–　　หาร

เครื่องหมายที่ใช้ในการคำนวณในแป้นพิมพ์ของไมโครคอมพิวเตอร์มีดังนี้

+　　บวก

–　　ลบ

*　　คูณ

/　　หาร

∧　　ยกกำลัง

ต่อไปนี้ เราจะลองใช้ไมโครคอมพิวเตอร์ ใช้คำนวณเลขแบบเครื่องคำนวณเลขธรรมดา

ลองกด A+B = (อย่าลืมกด RETURN) บนจอจะปรากฏ

? SYNTAX ERROR หมายความว่าเครื่องไม่เข้าใจคำสั่ง

ลองใหม่ กด 1+3 = เมื่อกด RETURN 1+3 = จะเลื่อนขึ้นไป เครื่องหมาย PROMPT และ CURSOR จะอยู่บรรทัดถัดมา

ลองกด PRINT 1+3 = RETURN เครื่องจะให้? SYNTAX ERROR อีก

ลองอีกที PRINT 1+3 เมื่อกด RETURN จะให้เลข 4 ที่บรรทัดถัดมา

ตอนนี้พอจะจับหลักได้แล้วว่า ให้พิมพ์คำว่า PRINT ก่อน แล้วจึงพิมพ์โจทย์ลงไป แล้วกด RETURN โดยไม่ต้องมีเครื่องหมายเท่ากับ

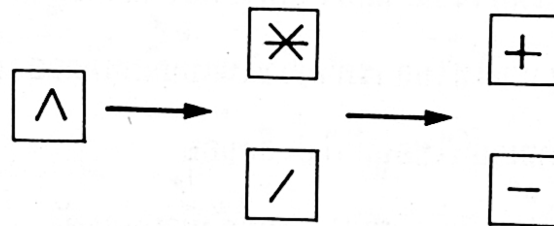
ลองใหม่ PRINT 2+5*4 จะได้ 22 เครื่องจะเอา 5 คูณกับ 4 ก่อน แล้วจึงเอาผลลัพธ์มาบวกกับ 2

ต่อไป PRINT 5Λ2*4 จะได้ 100 เครื่องจะเอา 5 ยกกำลัง 2 ได้ 25 แล้วไปคูณ 4

ต่อไป PRINT 60+4 - 3Λ2*5 จะได้ 19 เครื่องจะทำตามลำดับดังนี้

3Λ2	=	9
9*5	=	45
60+4	=	64
64-45	=	19

สรุป เครื่องจะทำงานตามลำดับเครื่องหมายดังนี้



เพื่อขจัดความสับสนในการคำนวณ เราอาจใช้วงเล็บแบ่งการทำงานเป็นตอนๆ ไป และวงเล็บอาจจะซ้อนๆ กันหลายๆ ชั้นได้ และเครื่องจะทำงานจากวงเล็บในออกมาข้างนอก

ตัวอย่าง PRINT (4*6) + (5/2) ผลลัพธ์ = 26.5

4*6	=	24
5/2	=	2.5
24+2.5	=	26.5

ตัวอย่าง PRINT((8+3) *2)/(4-2) ผลลัพธ์ = 11

8+3	=	11
11×2	=	22

4-2 = 2

22-2 = 11

ข้อควรจำ 1. พยายามใช้วงเล็บ

2. เมื่อใช้วงเล็บ จำนวนวงเล็บเปิดต้องเท่ากับวงเล็บปิด

สรุป ในบทที่ 3 นี้ ท่านได้เรียนรู้เรื่องดังนี้

1. รู้จักเครื่องหมาย PROMPT CHARACTER และ CURSOR
2. เครื่องจะทำงาน และให้คำตอบก็ต่อเมื่อกดแป้น คำว่า PRINT นำก่อน
3. ตัวที่ตามหลัง PRINT เป็นตัวเลข เครื่องจะทำงานให้ ถ้าเป็นตัวอักษร เครื่องจะไม่ทำงาน นอกจากตัวอักษรนั้นจะอยู่ในเครื่องหมายคำพูด
4. เครื่องหมายที่ใช้ในการคำนวณในแป้นพิมพ์ดีดของคอมพิวเตอร์ จะต่างกับเครื่องหมายที่ใช้อยู่ทั่วไปเล็กน้อย
5. ต้องกด แป้น RETURN ทุกครั้งที่จบบรรทัด
6. ถ้าในบรรทัดเดียวกันมีหลายเครื่องหมาย เครื่องจะทำงานตามลำดับ เครื่องหมาย ยกกำลัง คูณ หาร บวก ลบ
7. ถ้ามีวงเล็บ เครื่องจะทำงานจากวงเล็บในตามลำดับในข้อ 6 ออกมาวงเล็บนอก
8. ในบทนี้ การทำงานของคอมพิวเตอร์แบบนี้ เรียกว่า IMMEDIATE EXECUTION คือ เครื่องทำงานทันที เมื่อกด RETURN

แบบฝึกหัดบทที่ 3

จงหาผลลัพธ์ *คำตอบ*

3.1 4∧3 + 6*5 - 6/2 * 5 (79)

3.2 32/4 *2 - 4∧2 -5 (-5)

3.3 (3+ (12*3)) *3 (117)

3.4 4* (56- (6∧2)) * (32/8+1) (400)

ตัวแปร (VARIABLES)

4

4.1 **ตัวแปร** (VARIABLES) คือ ชื่อของหน่วยความจำในคอมพิวเตอร์ ซึ่งทำหน้าที่เก็บตัวเลข หรือค่าไว้ในตัวแปรเองและเราสามารถสั่งให้คอมพิวเตอร์ค้นหาตัวเลข หรือค่าที่เก็บไว้ในตัวแปรนั้น ด้วยการระบุชื่อตัวแปร

ลองนึกถึงภาพที่มีพนักงานคัดแยกเอกสารนั่งอยู่หน้าตู้ที่เป็นช่องเล็กๆ แต่ละช่องมีชื่อของเอกสารติดอยู่ เราต้องการให้เอาเอกสารฉบับหนึ่งใส่ในตู้ที่มีชื่อ X พนักงานก็จะมองหาตู้ที่มีชื่อ X อยู่ เมื่อพบก็เอาจดหมายใส่ในตู้นั้น ถ้าหาไม่ได้ก็จะเอาจดหมายนั้นใส่ในตู้ว่างๆ แล้วเขียนชื่อ X ติดไว้ ถ้าพนักงานพบว่าในตู้ X นั้น มีเอกสารฉบับอื่นอยู่แล้ว เขาจะเอาเอกสารเดิมออกมาทำลาย แล้วเอาเอกสารใหม่ใส่เข้าไป สมมติว่าเราสั่งให้พนักงานหยิบเอกสารที่มีอยู่ในตู้ X มาใช้ พนักงานจะทำสำเนาเอกสารนั้นแล้วเอาสำเนาที่เหมือนกันนั้นมาใช้ ส่วนจดหมายตัวจริงยังคงอยู่ในช่อง X ตามเดิม

ในภาษา BASIC จะมีตัวแปรที่เปรียบเสมือนตู้เอกสารดังกล่าวอยู่ประมาณ 900 ชื่อ โดยที่การเรียกชื่อตัวแปรเหล่านี้มีหลักเกณฑ์ดังนี้

ก. ชื่อตัวแปรจะต้องขึ้นต้นตัวแรกด้วยตัวอักษร

ข. ชื่ออาจจะเป็นตัวอักษรตัวเดียวโดดๆ เช่น A, B,... จนถึง Z

ค. ชื่ออาจจะมีตัวอักษรตามก็ได้ เช่น AA, DX, XM,...

ง. ชื่ออาจจะมีตัวเลขตามก็ได้ เช่น A1 ... A9, Z1 ... Z9

จ. การเขียนชื่ออาจจะนำด้วยอักษรตัวแรก แล้วตามด้วยตัวอักษรและตัวเลขปนกัน และจะยาวเท่าใดก็ได้ (แต่ไม่เกิน 238 ตัว) เช่น MA8-BC063KV..... แต่ตัวที่จะมีความหมายจริงๆ คือ 2 ตัวหน้า ตามตัวอย่างนี้ หากจะมีชื่อตัวแปรอีกตัวชื่อ MAXV65K8 ตัวแปรทั้ง 2 ตัวนี้ คือตัวเดียวกันคือ MA

ฉ. ชื่อของตัวแปรจะต้องไม่มีคำที่เป็นคำสงวนไว้เป็นคำสั่ง ดังมีรายชื่อคำอยู่ในภาคผนวกที่ 1

4.2 **การบรรจุตัวแปร** การที่จะเอาเลขจำนวนหนึ่งบรรจุเข้าไปในตัวแปรที่มีชื่อนั้น ในภาษา BASIC ใช้เครื่องหมายเท่ากับ (=) เช่น

A = 8 หมายถึงให้เอาเลขจำนวนหนึ่ง คือ 8 บรรจุเข้าไว้ในตัวแปร ซึ่งมีชื่อว่า A

ZEIKOISM = 364 หมายถึงให้เอาเลขจำนวน 364 บรรจุเข้าไว้ในตัวแปร ที่มีชื่อว่า ZE

เลขที่อยู่ในตัวแปรนี้ เราสามารถนำมาใช้เมื่อใดก็ได้ โดยการเรียกชื่อของตัวแปร

เปิดเครื่อง แล้วกด A = 8 (อย่าลืมปิดท้ายด้วย RETURN)

B = 2

การทำดังนี้เป็นการเอาค่า 8 ไปบรรจุไว้ในตัวแปรชื่อ A และค่า 2 บรรจุไว้ในตัวแปรชื่อ B

ต่อไป กด PRINT A + B คำตอบคือ 10 เครื่องจะเอาค่าในตัวแปร A ไปบวกกับค่าในตัวแปร B แล้วให้คำ

ตอบมา คือ 10

ลองกด C = 10 ให้ 10 บรรจุอยู่ในตัวแปร C

กด PRINT (A+B) * B/C คำตอบ คือ 2

4.3 **การเปลี่ยนเลขในตัวแปร** เราสามารถจะเปลี่ยนตัวเลขที่อยู่ในตัวแปรได้เสมอ โดยพิมพ์ตัวเลขใหม่เข้าไปแทนที่

ลองกด A = 4

PRINT A+B

ผลลัพธ์คือ 6 ตัวเลข 8 ที่เคยอยู่ใน A จะหายไป ตัวเลข 4 จะเข้ามาแทนที่ หากไม่เชื่อลองกด PRINT A ตัวเลข 4 จะปรากฏอยู่บรรทัดต่อมา

4.4 B = B + 1 ในวิชาพีชคณิต การเขียนดังนี้ จะเรียกว่าสมการ(ซึ่งเป็นสมการที่เป็นไปไม่ได้) แต่ในภาษา BASIC หมายความว่าให้เอาค่าในตัวแปรที่มีชื่อว่า B มาบวกเข้ากับ 1 แล้วเอาผลลัพธ์นี้ใส่กลับเข้าไปในตัวแปร B อีก

ดังนั้นเครื่องหมาย = ในภาษา BASIC จึงไม่เหมือนเครื่องหมาย = ในวิชาพีชคณิต

ตัวแปรตามที่กล่าวมาแล้วนั้น เป็นตัวแปรที่ใช้บรรจุตัวเลข เราเรียกว่า NUMERICAL VARIABLES มีตัวแปรอีกชนิดหนึ่งสามารถจะบรรจุตัวอักษรเข้าไปได้เป็นกลุ่ม ตัวแปรแบบนี้เรียกว่า STRING VARIABLES ซึ่งต่อไปจะเรียกว่า "ตัวแปรพันธะ" ลองพิมพ์ข้อความข้างล่างนี้ลงในเครื่อง

```
A$    =    "HELLO"

PRINT A$
```

เครื่องจะพิมพ์ HELLO ออกมา

เรื่องของตัวแปรพันธะ(STRING VARIABLES) ยังมีอีกมาก จะได้กล่าวละเอียดในบทที่ 15

สรุปข้อพึงจำในบทนี้

1. ตัวแปร(VARIABLE) เปรียบเสมือนช่องเก็บของมี 2 แบบ คือ ตัวแปรสำหรับเก็บตัวเลข(NUMERICAL VARIABLE) กับตัวแปรที่ใช้เก็บตัวอักษร(STRING VARIABLE)
2. ถ้าเอาตัวเลขหรืออักษร บรรจุเข้าในตัวแปรที่มีตัวเลข หรืออักษรอยู่แล้ว ของเดิมจะหายไป ของใหม่จะเข้าไปแทนที่ แต่เมื่อเรียกเอามาใช้ ของที่มีอยู่จะคงอยู่อย่างเดิม
3. ชื่อของตัวแปรจะยาวเท่าใดก็ได้(แต่ไม่เกิน 238 ตัว) แต่เครื่องจะยอมรับเพียง 2 ตัวแรก
4. ชื่อของตัวแปรจะต้องไม่ตรงกับ "คำต้องห้าม"

แบบฝึกหัดบทที่ 4

4.1 ในหนึ่งปีจะมี 365 วัน (YEAR = 365) และในสัปดาห์หนึ่งจะมี 7 วัน (WEEK = 7) มีวิธีใดที่จะสั่งให้เครื่องบอกว่าในหนึ่งปีจะมีกี่สัปดาห์

4.2 จงหาพื้นที่ของวงกลมที่มีรัศมี 5 หน่วย กำหนดให้ $\pi = 3.14159$

เฉลยแบบฝึกหัดบทที่ 4

4.1

```
YEAR   =   365
WEEK   =     7
PRINT YEAR/WEEK
52.1428572
```

4.2

```
RADIUS  =  5
AREA    =  3.14159 * RADIUS * RADIUS
PRINT AREA
78.5397501
```

5 โปรแกรม

5.1 ในบทที่แล้วเราได้เขียนคำสั่งให้เครื่องทำงาน โดยทำทีละบรรทัด การกระทำดังนี้เรียกว่า "ทำงานทันที"(IMMEDIATE EXECUTION) ในบทนี้เราจะเขียนคำสั่งหลายๆ คำสั่งแล้วนำมาจัดเรียงกันเป็นบรรทัด เมื่อหมดคำสั่งที่ต้องการแล้ว จึงสั่งให้เครื่องทำงานทีเดียว การกระทำดังนี้เรียกว่า DEFERRED EXECUTION หรือ "การหน่วงเวลา" คำสั่งที่นำมาเรียงกันเป็นชุดเช่นนี้ เรียกว่า "โปรแกรม"(PROGRAM)

ข้างล่างนี้ คือ โปรแกรม

```
10 A = 5
20 B = 10
30 C = A * B
40  PRINT C
50  END

]RUN
50
```

ข้อสังเกตุ

- หมายเลขข้างหน้า เรียกว่า หมายเลขบรรทัด(LINE NUMBER)
- จะต้องกดแป้น RETURN ท้ายบรรทัดทุกบรรทัด
- ลำดับของคำสั่งมีดังนี้

  บรรทัดที่ 10 ให้เลขจำนวน 5 อยู่ในตัวแปร A

  บรรทัดที่ 20 ให้เลขจำนวน 10 อยู่ในตัวแปร B

บรรทัดที่ 30 ให้นำตัวเลขในตัวแปร A
และตัวแปร B มาคูณกัน
เอาผลลัพธ์ที่ได้มาไว้ใน-
ตัวแปร C

บรรทัดที่ 40 ให้แสดงจำนวนเลขที่อยู่-
ในตัวแปร C ในจอภาพ

บรรทัดที่ 50 การบอกว่าจบโปรแกรม

- RUN คือ คำสั่งที่ให้เครื่องปฏิบัติตามโปรแกรมที่วางไว้ข้างต้น มีชื่อเรียกว่า SYSTEM COMMAND
- 50 คือ ผลลัพธ์ หรือค่าของตัวเลขที่อยู่ในตัวแปร C

ในบทต่อไปจะได้อธิบายรายละเอียดของส่วนต่างๆ ที่ประกอบขึ้นเป็นโปรแกรม

6 SYSTEM COMMANDS

6.1 ดังที่กล่าวไว้ใน 1.5 ว่า คำสั่งที่ใช้กับคอมพิวเตอร์ หรือ COMMANDS มีอยู่ 2 แบบ คือ SYSTEM COMMANDS และ PROGRAM COMMANDS

SYSTEM COMMANDS คือ คำสั่งที่สั่งให้เครื่องปฏิบัติตามโดยตรง ปกติคำสั่งแบบนี้จะไม่ได้เป็นส่วนหนึ่งของโปรแกรม(แต่มีบางคำที่สามารถใส่เข้าไปในโปรแกรม ซึ่งเรียกว่า PROGRAM COMMANDS ได้) SYSTEM COMMANDS ที่ใช้ในภาษา BASIC มีดังนี้

6.2 NEW เป็นคำสั่งที่สั่งให้เครื่องลบล้างความจำทั้งหมดที่มีอยู่ในเครื่องคอมพิวเตอร์ เพื่อจะเริ่มเขียนโปรแกรมเรื่องใหม่ มิฉะนั้นโปรแกรมใหม่กับเก่าจะปะปนกัน

6.3 LIST คือคำสั่งให้เครื่องแสดงโปรแกรมที่เขียนมาตั้งแต่ต้นจนจบโปรแกรม โดยแสดงออกทางจอภาพ ด้วยการเลื่อนขึ้นจากข้างล่างไปข้างบนหรือแสดงออกโดยการพิมพ์ด้วยเครื่องพิมพ์

LIST 50	เครื่องจะแสดงโปรแกรมเฉพาะบรรทัดที่ 50
LIST 40-100	เครื่องจะแสดงโปรแกรมตั้งแต่บรรทัดที่ 40 ถึงบรรทัดที่ 100

LIST-100 หรือ LIST,100 เครื่องจะแสดงโปรแกรมตั้งแต่ต้นจนถึง
บรรทัดที่ 100

LIST 60- หรือ LIST 60, เครื่องจะแสดงโปรแกรมตั้งแต่บรรทัดที่
60 จนจบ

6.4 **RUN** คือ คำสั่งที่สั่งให้เครื่องปฏิบัติตามโปรแกรมที่วางไว้ตั้งแต่ต้นจนจบโปรแกรม RUN 160 เครื่องจะปฏิบัติตามโปรแกรมตั้งแต่บรรทัดที่ 160 จนจบโปรแกรม

6.5 **HOME** โดยปกติแล้วเมื่อเริ่มเปิดเครื่อง เครื่องหมาย PROMPT และ CURSOR จะอยู่มุมซ้ายสุดของบรรทัดล่างสุดของจอภาพ และเมื่อเขียนโปรแกรมลงไปบรรทัดแรก เมื่อกด RETURN ข้อความบรรทัดนั้นจะเลื่อนขึ้นไปข้างบน ส่วนเครื่องหมาย PROMPT และ CURSOR จะอยู่ตำแหน่งเดิม เป็นเช่นนี้ทุกบรรทัด แต่เมื่อกด HOME ทุกอย่างที่อยู่บนจอภาพจะหายไป เครื่องหมาย PROMPT และ CURSOR จะปรากฏขึ้นใหม่ที่ส่วนบนซ้ายของจอภาพเมื่อเขียนโปรแกรม จะเริ่มจากบรรทัดบนลงมาข้างล่างทีละบรรทัดจนถึงบรรทัดล่างสุด หลังจากนี้ ภาพทั้งจอจะขยับขึ้น เหลือบรรทัดล่างไว้ให้เขียนโปรแกรมบรรทัดหลังสุด คำสั่ง HOME นี้สามารถบรรจุเข้าเป็นส่วนหนึ่งของโปรแกรมได้

6.6 **LOAD** คือ คำสั่งที่ให้เครื่องคอมพิวเตอร์รับโปรแกรมที่ส่งมาจากเทปบันทึกมาเก็บไว้ในเครื่อง ถ้าหากโปรแกรมอยู่ในแผ่น Diskette คำสั่ง LOAD จะต้องมีชื่อโปรแกรมด้วย สมมติว่าโปรแกรมนั้นมีชื่อว่า GOOD คำสั่งนั้นจะเป็นดังนี้ LOAD GOOD

6.7 SAVE คือคำสั่งที่ให้เครื่องคอมพิวเตอร์ส่งโปรแกรมในเครื่องเข้าบันทึกไว้ในเทปบันทึก ถ้าจะบันทึกโปรแกรมลงใน Diskette โดยใช้เครื่องบันทึกแผ่น คำสั่งนั้นจะต้องมีชื่อที่ตั้งขึ้น สมมติว่าจะตั้งชื่อโปรแกรมนั้นว่า GOOD คำสั่งจะต้องเป็นดังนี้ SAVE GOOD

6.8 [CTRL] C ขณะที่เครื่องรับคำสั่งให้ LIST หรือ RUN และเครื่องกำลังทำงานอยู่ ถ้าต้องการหยุดโปรแกรมนั้น ให้กดแป้นที่เขียนไว้ว่า CTRL แล้วกดค้างไว้ พร้อมกับกดแป้นอักษร C แล้วจึงปล่อยทั้ง 2 แป้น โปรแกรมจะหยุด

6.9 CONT เมื่อต้องการให้โปรแกรมที่ได้รับคำสั่งให้หยุดด้วย [CTRL] C ดำเนินต่อไป ให้กด [C] [O] [N] [T]

6.10 [CTRL] S ขณะที่เครื่องกำลัง LIST โปรแกรมอยู่ เมื่อต้องการหยุด ให้กด [CTRL] S โปรแกรมจะหยุด เมื่อจะให้โปรแกรมวิ่งต่อไป ให้กด [CTRL] S การกระทำดังนี้ จะสะดวกกว่าการใช้ [CTRL] C และ CONT ที่กล่าวไว้ใน 6.8 และ 6.9 การกด [CTRL] C ต่างกับการกด [CTRL] S ที่ว่า [CTRL] C เป็นการหยุดการทำงานของเครื่อง ส่วนการกด [CTRL] S เป็นการหยุดการ LIST ชั่วคราว

6.11 TRACE เมื่อให้คำสั่งนี้แก่เครื่องก่อน จะ RUN หรือดำเนินการตามโปรแกรม เครื่องจะแสดงออกทางภาพว่า เครื่องกำลังดำเนินไปถึงโปรแกรมบรรทัดไหน เพื่อประโยชน์ในการตรวจสอบ

6.12 NOTRACE เป็นคำสั่งที่ยกเลิกคำสั่ง TRACE ตาม 6.11

6.13 DEL คือคำสั่งที่ให้ลบล้างโปรแกรม ซึ่งจะต้องตามด้วยหมายเลขบรรทัดเสมอ เช่น DEL 30 , 30 ให้ลบล้างโปรแกรมบรรทัดที่ 30 หากตามด้วยหมายเลขบรรทัด 2 กลุ่ม เช่น DEL 30–40 หมายถึง ลบล้างคำสั่งตั้งแต่บรรทัด 30 ไปจนถึง 40

ในระยะแรกผู้เริ่มศึกษาอาจจะไม่เข้าใจในคำสั่งเหล่านี้ แต่เมื่อศึกษาตามลำดับ และใช้คำสั่งเหล่านี้บ่อยครั้ง จะเข้าใจความหมายได้ดี

7

LINE NUMBER

7.1 LINE NUMBER หรือ STATEMENT NUMBER เป็นสิ่งที่จำเป็นสำหรับการเขียนโปรแกรมภาษา BASIC LINE NUMBER ก็คือหมายเลขสำหรับบรรทัดของโปรแกรมที่เราต้องการให้เครื่องปฏิบัติตามโดยเริ่มจากเลขน้อยไปเลขมากตามลำดับ

สมมติว่าจะขับรถไปซื้อของ มีลำดับการกระทำโดยย่อตามลำดับดังนี้ ขึ้นรถ, ขับรถ, ลงจากรถ เราอาจจะใส่หมายเลขการกระทำดังนี้

1. ขึ้นรถ
2. ขับรถ
3. ลงจากรถ

หากให้หมายเลขลำดับสับสนผิดไปจากข้อเท็จจริงดังนี้แล้วสั่งให้คอมพิวเตอร์ทำงาน เช่น

1. ขับรถ
2. ขึ้นรถ
3. ลงจากรถ

คอมพิวเตอร์จะปฏิเสธที่ทำตาม แต่ถ้าลำดับหมายเลขตรงกับข้อเท็จจริง แต่การเขียนโปรแกรมสลับที่กันดังนี้

2. ขับรถ

1. ขึ้นรถ

3. ลงจากรถ

แม้ว่าการพิมพ์ลำดับหมายเลขบรรทัดจะไม่เรียงกัน แต่เมื่อนำมาจัดเรียงกัน จะไม่ผิดข้อเท็จจริง เครื่องจะทำงานตามหมายเลขบรรทัดจากน้อยไปหามาก

ลองพิมพ์โปรแกรมต่อไปนี้ (อย่าลืมกดแป้น RETURN ทุกครั้งที่จบบรรทัด)

```
 2 PRINT"MORNING"

 1 PRINT"GOOD"

 3 END

]RUN
GOOD
MORNING
```

โปรแกรมนี้เราตั้งใจใส่หมายเลขให้สับสน เมื่อเขียนโปรแกรม 3 บรรทัด นับว่าโปรแกรมนี้สมบูรณ์แล้ว

กดแป้น R U N สั่งให้เครื่องทำงาน เครื่องจะทำงานเริ่มจากบรรทัดที่ 1 โดยแสดงคำว่า GOOD ตามที่สั่งไว้ ต่อมาเมื่อถึงบรรทัดที่ 2 จะพิมพ์คำว่า MORNING ตามที่สั่งไว้ เมื่อผ่านมาถึงบรรทัดที่ 3 เครื่องจะหยุด เพราะได้สั่งไว้ว่าหมดโปรแกรมแล้ว ถ้าจะเริ่มเดินโปรแกรมนี้ใหม่ก็กดเครื่องก็จะทำงานอีกครั้ง

```
]LIST

1   PRINT "GOOD"
2   PRINT "MORNING"
3   END
```

ลองกด L I S T ดู เครื่องจะแสดงโปรแกรมทั้ง 3 บรรทัดให้ดู

7.2 ในเครื่องคอมพิวเตอร์ APPLE II PLUS สามารถจะเขียนโปรแกรมได้ตั้งแต่บรรทัดหมายเลข 0 จนถึง 63999 บรรทัด วิธีที่นิยมกันนั้น จะกำหนดหมายเลขบรรทัดเป็นจำนวน 10 เช่น เริ่มที่ 10 แล้ว 20, 30, 40 ตามลำดับ จนจบโปรแกรม หรืออาจจะเริ่มที่ 110, 120, 130, ... ก็ได้ การกระทำดังนี้จะเป็นการง่ายที่จะแชกบรรทัดใด บรรทัดหนึ่งของโปรแกรมลงไปได้โดยไม่ต้องเขียนใหม่

ข้อปลีกย่อยในการเขียนโปรแกรม

8

8.1 แป้น [←] จะเป็นแป้นสำหรับแก้คำผิด เมื่อกด [←] CURSOR จะเลื่อนกลับไปทางซ้าย แล้วกระพริบอยู่ที่ตัวอักษรตัวสุดท้ายที่พิมพ์ลงไป ถ้าต้องการแก้ก็ให้พิมพ์คำที่ต้องการทับไป

ตัวอย่าง

```
10 PRINT"DE□
```

กด [←] บนจอจะเป็น

```
10 PRINT"DE
```

กด A จะเป็น

```
10 PRINT"DA□
```

พิมพ์ต่อจนจบบรรทัด และกด [RETURN] จะเป็นดังนี้

```
10 PRINT"DANG"
]□
```

8.2 เมื่อรู้ว่าพิมพ์ตัวใดตัวหนึ่งผิดหลังจากที่พิมพ์ตัวอื่นตามหลังมาหลายตัวแล้ว แต่ยังไม่ทันกดแป้น [RETURN]

```
]10 PRINT"GOAD MORNING DANG"□
```

ตัว A ในคำว่า GOAD ควรจะเป็นตัว O ให้ทำดังนี้ กด ← ซ้ำหลายๆ ที จนกระทั่ง CURSOR เลื่อนไปทางซ้ายทับตัวหนังสือที่จะแก้

```
10 PRINT"GOAD MORNING DANG"
```

กดตัว O ไปแทนที่ตัว A CURSOR จะเลื่อนไปทางขวามือทับตัว D

```
10 PRINT"GOOD MORNING DANG"
```

ต่อไปกดแป้น → CURSOR จะเลื่อนไป 1 ตัว กด → ซ้ำหลายๆ ทีจนกระทั่ง CURSOR เลื่อนไปจนสุดบรรทัด แล้วจึงกด RETURN

```
10 PRINT"GOOD MORNING DANG"
```

การเลื่อน CURSOR ไปทางขวานี้ มีลักษณะพิเศษอยู่อย่างหนึ่ง คือ ไม่ว่า CURSOR ทับอักษรตัวใด อักษรตัวนั้นจะเข้าหน่วยความจำเสมือนได้พิมพ์อักษรนั้นเข้าไป แป้นนี้จึงเรียกว่า RETYPE KEY

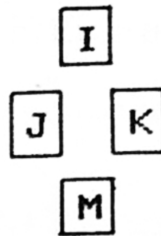
ถ้าจะให้การเลื่อน CURSOR ไปทางซ้ายหรือขวาเป็นไปอย่างรวดเร็ว ให้กดแป้น REPT ค้างไว้ แล้วกด ← หรือ → ตามที่ต้องการ CURSOR จะวิ่งไปอย่างเร็วไม่เสียเวลา

8.3 การแก้คำผิดหลังจากที่พิมพ์จนหมดบรรทัด และกด RETURN แล้วมีวิธีตามตัวอย่างนี้

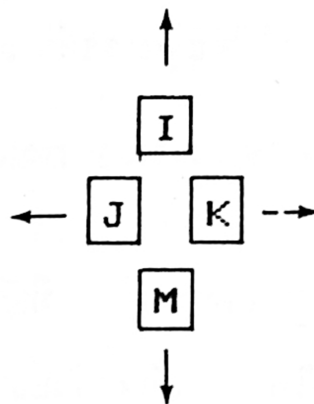
```
]10 PRINT"GOAD MORNING DANG"

]
```

ในแป้นพิมพ์จะมีอักษรอยู่ 4 ตัว ที่เรียงกันดังนี้



แป้นทั้ง 4 นี้มีลักษณะพิเศษที่ว่า เมื่อกดแป้น ESC ครั้งเดียวแล้วปล่อยมือต่อไปเมื่อพิมพ์อักษรใดใน 4 ตัวนี้ CURSOR จะเลื่อนไปได้ 4 ทิศทาง โดยไม่กระทบกระเทือนกับอักษรใดที่ได้พิมพ์ไปแล้ว ทิศทางของการเลื่อนเขียนได้ดังนี้



และการเคลื่อนไหวโดยไม่กระทบกระเทือนตัวอักษรที่พิมพ์ไปแล้วนี้ จะสิ้นสุดลงเมื่อพิมพ์แป้นตัวอักษรอื่นใดนอกจาก 4 ตัว ดังกล่าวลงไป

ตามตัวอย่าง เราจะแก้คำผิดในบรรทัดที่เราได้พิมพ์สิ้นสุดไปแล้ว (กด RETURN แล้ว) ให้ทำดังนี้

กด ESC หนึ่งครั้ง กด I หนึ่งครั้ง CURSOR จะเลื่อนขึ้นไปข้างบนหนึ่งบรรทัด ให้กด I ไปจนถึงบรรทัดที่ต้องการ หลังจากนั้นก็กด J ให้ CURSOR เลื่อนไปทางซ้ายจนไปทับตัวเลขตัวแรกของบรรทัด แล้วกด SPACE BAR หนึ่งครั้ง จากนั้นก็กด → จนกระทั่ง CURSOR ไปทับตัวที่ต้องการจะแก้แล้วจึงพิมพ์ตัวที่ถูกทับลงไป แล้วกด → ไปจน CURSOR เลื่อนไปหมด

บรรทัด แล้วกด RETURN

ด้วยวิธีนี้เราสามารถย้อนขึ้นไปแก้บรรทัดต้นๆ ห่างออกไปจากบรรทัดที่กำลังพิมพ์อยู่ได้

8.4 ถ้าพิมพ์ดังนี้ 50 PRINT RETURN เมื่อโปรแกรมเดินถึงบรรทัดนี้ เครื่องจะไม่ปฏิบัติอย่างใดนอกจากจะเว้นบรรทัดให้

8.5 เมื่ออยากจะเปลี่ยนโปรแกรมใหม่ทั้งบรรทัดโดยไม่อยากแก้ของเดิม ให้พิมพ์โปรแกรมใหม่โดยใช้หมายเลขบรรทัดเดิม เมื่อพิมพ์เสร็จ กด RETURN โปรแกรมนั้นจะเข้าไปแทนที่โปรแกรมบรรทัดเดิม

ตัวอย่าง

```
130 PRINT"HELLO DANG"
```

พิมพ์ใหม่

```
130 PRINT"HELLO DUM"
```

โปรแกรมบรรทัดที่พิมพ์ครั้งหลังจะเข้าไปแทนที่ที่พิมพ์ไว้แต่เดิม

8.6 เมื่ออยากจะยกเลิกบรรทัดใดไปเลยโดยไม่มีการทดแทน สมมติว่าจะเลิกบรรทัด 30 ดังที่กล่าวข้างบน ทำให้ 2 วิธีคือ พิมพ์ DEL 30 ,30 RETURN หรือ 30 RETURN หรือ 30 CTRL X บรรทัดที่ 30 จะถูกลบออก(เมื่อกด 30 CTRL X บนจอภาพ จะปรากฏดังนี้ 30/)

8.7 ต้องการที่จะหมายเหตุเป็นการบันทึกความจำไว้ในโปรแกรมซึ่งจำเป็นอย่างมากสำหรับโปรแกรมต่างๆ ทำได้โดยใช้คำว่า REM (ย่อมาจาก REMARK)

หลังหมายเลขบรรทัด แล้วตามด้วยข้อความใดก็ได้ เช่น

```
10 REM THIS IS MY FIRST PROGRAM
```

บรรทัดที่มีคำว่า REM ตามท้ายหมายเลขบรรทัดเครื่องจะเลยไป ไม่ทำงานในบรรทัดนั้น แต่เมื่อ LIST โปรแกรมนี้ออกมาดู บรรทัดนี้จะปรากฏอยู่

8.8 คำว่า PRINT จะใช้มากในโปรแกรม เราอาจจะใช้เครื่องหมายคำถาม (?) แทนคำว่า PRINT ได้

```
50 PRINT"DANG"
```

อาจจะพิมพ์ดังนี้

```
50 ? "DANG"
```

8.9 บางครั้งเราอาจจะใช้คำสั่งหลายคำสั่งไว้ในบรรทัดเดียวกัน โดยใช้เครื่องหมาย Colon (:) เป็นตัวกั้น

```
30 PRINT"DANG":PRINT"TONG"
```

8.10 พิมพ์ข้อความต่อไปนี้ลงในเครื่อง

```
10 REM THIS IS MY FIRST PROGRAM
20 PRINT"BASIC PROGRAMMING"
30 ? "IS EASY"
```

ปฏิบัติตามคำสั่งต่อไปนี้ แล้วสังเกตผลที่ออกมาทางจอ (อย่าลืมกด RETURN)

- กด R U N เครื่องจะพิมพ์คำที่อยู่ในเครื่องหมายคำพูด

ของบรรทัดที่ 20 และ 30

- กด L I S T — เครื่องจะแสดงเหมือนโปรแกรมที่เขียนมา สังเกตบรรทัดที่ 30 เครื่องหมาย ? จะหายไป มีคำ PRINT มาแทน

- กด 30 PRINT "IS NOT HARD" [RETURN]

  แล้วกด LIST จะเห็นว่าบรรทัดที่ 30 จะเปลี่ยนใหม่

- กด RUN — สังเกตบรรทัดสุดท้าย

- กด LIST แล้วทำดังนี้ตามลำดับ

  กด ESC — หนึ่งครั้ง

  กด I — 2 ครั้ง

  กด J — 1 ครั้ง

  กด SPACE BAR 1 ครั้ง

  กด [→] — จนกระทั่ง CURSOR เลื่อนไปอยู่ตรงตัว N ของคำว่า NOT

  กด คำ EASY" ทับลงไป แล้วกด [RETURN]

- กด LIST — สังเกตดูว่าบรรทัดที่ 30 จะเปลี่ยนไป

- กด RUN — สังเกตบรรทัดสุดท้าย

- กด LIST — แล้วลองเปลี่ยนคำว่า EASY เป็น NOT HARD

PRINT

9

9.1 PRINT เป็น SYSTEM COMMAND ที่สั่งให้เครื่องแสดงค่า หรืออักษรที่เราต้องการออกมาบนจอภาพ หรือถ้ามีเครื่องพิมพ์ก็ให้พิมพ์ออกมา คำสั่ง PRINT คือ การเสนอข่าวสารจากโปรแกรมให้แสดงแก่โลกภายนอก

9.2 ในบทที่ 8 เราได้เรียนรู้แล้วว่า ถ้าพิมพ์คำสั่ง PRINT หลังหมายเลขบรรทัดแล้วไม่มีตัวอื่นตาม เครื่องจะทำการเว้นบรรทัดให้ และคำว่า PRINT อาจใช้ ? แทนได้

9.3 สังเกตโปรแกรมนี้

```
 10 A=10
 20 PRINT A
 30 END
]RUN
10
```

บรรทัดที่ 10 เราบรรจุข้อมูลคือ เลข 10 เข้าในตัวแปร A

บรรทัดที่ 20 เราสั่งให้เครื่องแสดงค่าในตัวแปร A

เมื่อ RUN ออกมา เครื่องจะแสดงค่า A คือ 10 ออกมา

ดูโปรแกรมนี้

```
10 PRINT"GOOD MORNING"
20 PRINT"DANG"
```

บรรทัดที่ 10 และ 20 เราสั่งให้พิมพ์คำว่า GOOD MORNING และ DANG

```
30 END

]RUN
GOOD MORNING
DANG
```

ตามลำดับ

สังเกตว่าโปรแกรมนี้ต่างกับโปรแกรมแรกที่ว่า ตัวอักษรจะต้องอยู่ในเครื่องหมายคำพูด ซึ่งเรื่องนี้เราจะได้เรียนรู้ในเรื่อง STRING VARIABLES ในบทที่ 15 ต่อไป

9.4

```
10 PRINT 10

20 PRINT 20

30 PRINT 30

]RUN
10
20
30
```

ในโปรแกรมนี้ คำตามหลัง PRINT เป็นตัวเลข เครื่องจะพิมพ์ตัวเลขนั้นในบรรทัดถัดมาเรียงกัน คำตอบละบรรทัด

9.5

```
NEW

10 PRINT 10;20;30;40
]RUN
10203040
```

ใน PRINT บรรทัดเดียว เราจับตัวเลขมาเรียงกันคั่นด้วยเครื่องหมาย ; (Semi Colon) เครื่องจะแสดงออกมาในรูปที่ตัวเลขเรียงติดต่อกัน

9.6

```
 NEW

 10 PRINT 10,20,30

]RUN
10              20              30
```

หากตัวเลขในบรรทัดเดียวกันนั้น มีเครื่องหมาย Comma (,) คั่น เครื่องจะพิมพ์ทิ้งระยะกันเป็นช่วง ช่วงละ 16 ระยะตัวอักษร(Character)

ข้อพึงสังเกต โปรแกรมในข้อ 9.5 และ 9.6 เป็นการพิมพ์บรรทัดที่ 10 บรรทัดเดียว แต่ในข้อ 9.4 มีบรรทัดที่ 20 และ 30 อยู่ เท่ากับเราแก้บรรทัดที่ 10 บรรทัดเดียว วิธีแก้ไขก็คือ กด NEW เป็นการลบล้างโปรแกรมใดก็ตามที่มีอยู่ในเครื่องให้หมดก่อนที่จะเขียนโปรแกรมใหม่ใน 9.5 และ 9.6

9.7

```
10  REM  PROGRAM #9-1
20  HOME
30  PRINT "EXPENSE FOR JAN AND FE
    B"
40  PRINT " ","JAN","FEB"
50  PRINT " ","---","---"
60  PRINT "GAS",350,300
70  PRINT "TELEPHONE",80,95
80  PRINT "ELECTRICITY",250,280
90  PRINT "HOUSE",,2000
100  PRINT
110  PRINT "TOTAL",350 + 80 + 250
     ,300 + 95 + 280 + 2000
120  END

]RUN
EXPENSE FOR JAN AND FEB
                JAN             FEB
                ---             ---
GAS             350             300
TELEPHONE       80              95
```

ELECTRICITY	250	280
HOUSE		2000
TOTAL	680	2675

บรรทัดที่ 10 เป็นชื่อโปรแกรมที่จะไม่ได้แสดงออกทางจอภาพ

บรรทัดที่ 20 เป็นการทำให้จอภาพว่างและให้ข้อมูลที่แสดงต่อไป เริ่มจากส่วนบนของจอ

บรรทัดที่ 30 แสดงข้อความธรรมดา

บรรทัดที่ 40 ในเครื่องหมายคำพูดอันแรกไม่มีข้อความใด แต่มีเครื่องหมายคอมม่า (,) ตาม ดังนั้นข้อความในเครื่องหมายคำพูดต่อไปจะแสดงออกโดยเว้นไป 16 ตัวอักษรเช่นเดียวกับข้อความในเครื่องหมายคำพูด ตัวที่ 3 "FEB" จะเว้นไปอีก 16 ตัวอักษร (นับจากตัว J)

บรรทัดที่ 50 เป็นการขีดเส้นใต้

บรรทัดที่ 60, 70, 80 และ 90 เป็นการพิมพ์รายการโดยให้แต่ละคอลัมน์อยู่ในแนวตรงกัน สังเกตดูบรรทัดที่ 90 มีคอมม่า 2 ตัวตามกัน แบบนี้เครื่องจะเว้นคอลัมน์ที่ 2 ไปพิมพ์คอลัมน์ที่ 3 เลย

บรรทัดที่ 100 จะเป็นการรวมตัวเลข คำสั่ง + จะบวกตัวเลขให้เป็นผลลัพธ์ออกมาเลย

สรุป เครื่องหมายที่ตามหลังตัวเลข หรือข้อความที่ตามหลังคำสั่ง PRINT จะเป็นการกำหนดให้การพิมพ์เป็นไปในลักษณะต่างๆ ดังนี้

พิมพ์ติดต่อกันไป

พิมพ์เว้นเป็นช่วง โดยที่ระยะห่างจากตัวหน้าของช่วงหลัง ห่างจากตัวหน้าของช่วงแรก 16 ตัวอักษร หากช่วงแรกมีอักษร หรือตัวเลขมากกว่า 16 ตัวอักษร ช่วงหลังจะข้ามไปในช่วงที่ถัดไป

แบบฝึกหัดบทที่ 9

9.1 จงเขียนตัวเลขเหล่านี้เป็น 3 คอลัมน์เรียงกัน

13560	18950	25430	15320	69450
13680	36470	86760	43230	

และจงรวมเลขแต่ละคอลัมน์ นำไปพิมพ์ในบรรทัดสุดท้าย โดยมีการเว้นบรรทัดก่อนหนึ่งบรรทัด

เฉลยแบบฝึกหัดบทที่ 9

9.1

```
5  REM  S9.1
10  PRINT 13560,18950,25430
20  PRINT 15320,69450,13680
30  PRINT 36470,86760,43230
40  PRINT
50  PRINT 13560 + 15320 + 36470,1
     8950 + 69450 + 86760,25430 +
     13680 + 43230

]RUN
13560               18950               25430
15320               69450               13680
36470               86760               43230

65350               175160              82340
```

10

INPUT และ GET

10.1 ในการเขียนโปรแกรม บางครั้งเราจะต้องหยุดโปรแกรม เพื่อที่จะส่งข้อมูลให้แก่เครื่อง คำสั่งดังกล่าวนี้เรียกว่า INPUT STATEMENT เมื่อเครื่องดำเนินตามโปรแกรมไปถึงบรรทัดนี้เครื่องจะหยุด แล้วแสดงเครื่องหมายคำถาม เมื่อป้อนข้อมูลให้แก่เครื่องทางแป้นพิมพ์ดีด แล้วกด RETURN เครื่องจะรับข้อมูลเข้า และดำเนินต่อไป

10.2 ดูโปรแกรมนี้

```
10  REM   PROGRAM #10-1
20  INPUT A
30  INPUT B
40 C = A * B
50  PRINT C
60  END


]RUN
?50
?60
3000
```

เมื่อเครื่องดำเนินมาถึงบรรทัดที่ 20 จะหยุด และแสดงเครื่องหมายคำถาม เราต้องให้ข้อมูลโดยการกดแป้น แล้วกด RETURN เครื่องจะมาถึงบรรทัดที่ 30 หยุดคอยข้อมูลอีก เมื่อให้ข้อมูลแก่บรรทัดที่ 30 และกด RETURN แล้วเครื่องจะเดินไปจนจบโปรแกรม

10.3 ในโปรแกรมข้างบน ถ้าเปลี่ยนคำว่า INPUT ในบรรทัดที่ 20 และ 30 เป็น GET ดังนี้

```
]20 GET A
]30 GET B
```

RUN โปรแกรมนี้ดู จะสังเกตเห็นข้อแตกต่างจากโปรแกรมข้างบน 2

ประการคือประการแรกเมื่อเครื่องคอยข้อมูล จะไม่แสดงเครื่องหมายคำถามและข้อมูลจะป้อนได้เพียงตัวเดียว เครื่องจะดำเนินไปเองโดยไม่ต้องกด [RETURN]

```
10   REM    PROGRAM #10-1
20   GET  A
30   GET  B
40 C = A * B
50   PRINT C
60   END

]RUN
30
```

ในกรณีนี้ ถ้าเราพยายามป้อนข้อมูลให้ A = 50 และ B = 60 จะได้เพียง A = 5, B = 6 เท่านั้น และ CURSOR จะไม่แสดงอาการรับรู้ว่ามามีตัวเลขเข้า จนกว่าการป้อนข้อมูลครบจึงจะให้ผลออกมา

10.4 หลังคำสั่ง INPUT เราสามารถตามด้วยตัวอักษรที่อยู่ในเครื่องหมายคำพูดได้

```
10   REM    PROGRAM #10-2
20   PRINT "AREA OF TRIANGLE"
30   INPUT "BASE LENGTH= ";A
40   INPUT "HEIGHT= ";B
50   PRINT
60   PRINT "AREA= ";A * B / 2
70   END

]RUN
AREA OF TRIANGLE
BASE LENGTH= 5
HEIGHT= 6

AREA= 15
```

RUN โปรแกรมนี้ดู แล้วให้ค่าของ BASE LENGTH และ HEIGHT จะได้คำตอบเป็น AREA เมื่อจะเดินโปรแกรมใหม่ ให้พิมพ์ RUN อีกครั้ง

สังเกตดูบรรทัดที่ 30 และ 40 ตัวอักษรที่อยู่ในเครื่องหมายคำพูด และตัวแปรมีเครื่องหมาย ; คั่นอยู่

สรุป เมื่อเครื่องพบคำว่า INPUT หรือ GET เครื่องจะหยุดเพื่อคอยการป้อนข้อมูล ถ้าเป็น INPUT เมื่อป้อนข้อมูลไปแล้ว จะต้องกด RETURN ถ้าเป็น GET จะป้อนข้อมูลได้เพียง 1 ตัวเท่านั้น และไม่ต้องกด RETURN

แบบฝึกหัดบทที่ 10

10.1 จงเขียนโปรแกรม บอก อายุ น้ำหนัก และความสูงของท่าน โดยระบุให้ชัดเจน เช่น

```
I AM 25 YEARS OLD.
MY WEIGHT IS 50 KG.
I AM 1.58 M. TALL.
```

โดยใช้คำสั่ง INPUT ให้ป้อนตัวเลขได้

เฉลยแบบฝึกหัดบทที่ 10

10.1

```
5  REM  S10.1
10  INPUT "AGE";A
20  INPUT "WEIGHT";W
30  INPUT "HEIGHT";H
35  HOME
40  PRINT "I AM  ";A;" YEARS OLD"
50  PRINT "MY WEIGHT IS  ";W;" KG
   ."
60  PRINT "I AM  ";H;"  M. TALL"

]RUN
AGE15
WEIGHT38
HEIGHT1.50
I AM  15 YEARS OLD
MY WEIGHT IS  38 KG.
I AM  1.5  M. TALL
```

11

GOTO

11.1 ตามที่ได้กล่าวไว้ในบทที่ 7 ว่า เครื่องจะทำงานตามโปรแกรม โดยเริ่มจากบรรทัดหมายเลขน้อยไปตามบรรทัดหมายเลขมาก ตามลำดับ แต่ในบางครั้ง เราอาจจะต้องการให้เครื่องกระโดดไปทำงานที่บรรทัดอื่นที่เราต้องการก็ได้ การกระทำดังนี้เรียกว่า JUMP หรือ BRANCHING มีอยู่ 2 แบบคือ

1. การกระโดดไปทำงานที่บรรทัดที่ต้องการ โดยไม่มีข้อแม้ใดๆ (Unconditional Jump) คำสั่งที่ใช้ คือ GOTO
2. การกระโดดไปทำงานที่บรรทัดที่ต้องการ โดยมีข้อแม้ตามเครื่องหมายเปรียบเทียบ(Conditional Jump) โดยใช้คำสั่ง IF...THEN...(ดูบทที่ 12)

11.2 พิมพ์โปรแกรมข้างล่างนี้

```
10  REM   PROGRAM #11-1
20  PRINT "EXAMPLE OF GOTO"
30  GOTO 60
40  PRINT "ONE IS CALLED CONDITIO
     NAL JUMP"
50  GOTO 80
60  PRINT "THERE ARE TWO KINDS OF
      JUMPS"
70  GOTO 40
80  PRINT "ANOTHER IS UNCONDITION
     AL JUMP"
90  END
```

```
]RUN
EXAMPLE OF GOTO
THERE ARE TWO KINDS OF JUMPS
ONE IS CALLED CONDITIONAL JUMP
ANOTHER IS UNCONDITIONAL JUMP
```

โปรแกรมนี้ แสดงว่าเราจะสั่งให้เครื่องทำงานที่บรรทัดไหนก็ได้ โดยใช้คำสั่ง GOTO แล้วตามด้วยหมายเลขบรรทัดที่สั่งให้เครื่องไปกระทำ เมื่อเครื่องทำงานผ่านบรรทัดที่ 30 แล้ว ควรจะไปบรรทัดที่ 40 แต่เราสั่งให้ไปทำงานที่บรรทัดที่ 60 ก่อน ลองเปรียบเทียบผลของการ RUN กับโปรแกรมที่เขียนไว้

11.3 พิมพ์โปรแกรมข้างล่างนี้

```
10  REM   PROGRAM #11-2
20  PRINT "HOW NICE YOU ARE !"
30  GOTO 20

]RUN
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !
HOW NICE YOU ARE !

BREAK IN 20
```

โปรแกรมนี้ เมื่อ RUN แล้วจะไม่หยุด จะพิมพ์ HOW NICE YOU ARE! ไปเรื่อยๆ เพราะโปรแกรมหมุนอยู่ระหว่างบรรทัดที่ 20 และ 30 วิธีจะให้หยุด

ต้องกด CTRL C การที่โปรแกรมกลับไปกลับมาระหว่าง 2 บรรทัดนี้ เราเรียกว่า "เข้าวงรอบ" "(LOOP)"

11.4 นำโปรแกรม 10-2 มาพิมพ์อีกที แต่เปลี่ยนบรรทัด 70 ดังนี้

```
70 GOTO 20
```

แล้ว RUN ดู จะพบว่าแทนที่โปรแกรมจะจบในบรรทัดที่ 70 กลับย้อนไปที่บรรทัดที่ 20 ดังนั้นเราสามารถจะใช้โปรแกรมเดียวกันนี้หาพื้นที่ของรูปสามเหลี่ยมรูปใหม่ โดยป้อนข้อมูลชุดใหม่เข้าไป

สรุป คำสั่ง GOTO เป็นคำสั่งที่ให้โปรแกรมไปทำงานตามบรรทัดที่กำหนด โดยไม่ต้องดำเนินตามลำดับหมายเลขบรรทัด

แบบฝึกหัดบทที่ 11

11.1 จงเขียนโปรแกรมถามปีเกิด แล้วให้คำตอบเป็นอายุ และให้ใช้โปรแกรมนี้ได้กับหลายคน

เฉลยแบบฝึกหัดบทที่ 11

11.1

```
5  REM  S11.1
10  HOME
20  INPUT "IN WHAT YEAR WERE YOU
     BORN ?  ";B
30  INPUT "WHAT YEAR IS IT NOW ?
     ";N
40  PRINT "YOU ARE ";N - B;" YEAR
     S OLD"
50  PRINT
60  GOTO 20

]RUN
IN WHAT YEAR WERE YOU BORN ?  1968
WHAT YEAR IS IT NOW ? 1983
YOU ARE 15 YEARS OLD

IN WHAT YEAR WERE YOU BORN ?  1932
WHAT YEAR IS IT NOW ? 1983
YOU ARE 51 YEARS OLD
```

12

IF.... THEN....

12.1 ฉลองกับสมควรไปรับประทานอาหารกลางวันด้วยกัน ถ้า (IF) ฉลองมีเงินในกระเป๋ามากกว่า (>) สมควร แล้ว (THEN) ฉลองจะเป็นผู้ออกเงินค่าอาหาร ถ้า (IF) ฉลองมีเงินน้อยกว่า (<) สมควร แล้ว (THEN) สมควรจะเป็นผู้ออก ถ้า (IF) ทั้งสองคนมีเงินเท่ากัน (=) แล้ว (THEN) ต่างคนต่างออก

นี่คือ ตัวอย่างง่ายๆ ของคำสั่ง IF ... THEN ...

12.2 80 <u>IF</u> A > B THEN C = A + B

ตามตัวอย่างนี้ เมื่อโปรแกรมดำเนินถึงบรรทัดที่ 80 พบคำว่า IF จะพิจารณาการเปรียบเทียบ หากพบว่าการเปรียบเทียบนั้นเป็นจริง (ตามตัวอย่าง หาก A มากกว่า B จริง) เครื่องจะปฏิบัติคำสั่งที่ตามหลังคำว่า THEN (คือเอา A มาบวกกับ B แล้วนำไปบรรจุไว้ในตัวแปร C) หากว่าการเปรียบเทียบนั้นไม่จริง (คือ A เท่ากับ B หรือ A น้อยกว่า B) แล้วเครื่องจะหยุดไม่ดำเนินต่อไปในบรรทัดนั้น แต่จะลงมาบรรทัดล่างต่อไปทันที

12.3 ถ้าเราเขียนเป็นรูปแบบฟอร์มจะเขียนได้ดังนี้

80 <u>IF</u> ถ้าเป็นจริง THEN ทำตามคำสั่งต่อไปนี้ → ....

80 <u>IF</u> ถ้าไม่เป็นจริง┐ THEN

└→90 .....

12.4 พิจารณาโปรแกรมต่อไปนี้อย่างละเอียด

```
10  REM   PROGRAM #12-1
20  INPUT "GUESS NUMBER FROM 1 TO
      10 ";A
30  REM  MY NUMBER IS 7
40  IF A = 7 THEN  GOTO 70
50  IF A < 7 THEN  GOTO 90
60  IF A > 7 THEN  GOTO 110
70  PRINT "YOU ARE RIGHT"
80  END
90  PRINT "TOO LOW, TRY AGAIN"
100  GOTO 20
110  PRINT "TOO HIGH, TRY AGAIN"
120  GOTO 20

]RUN
GUESS NUMBER FROM 1 TO 10 3
TOO LOW, TRY AGAIN
GUESS NUMBER FROM 1 TO 10 8
TOO HIGH, TRY AGAIN
GUESS NUMBER FROM 1 TO 10 7
YOU ARE RIGHT
```

โปรแกรมนี้ เครื่องจะให้เราเดาตัวเลขจาก 1 ถึง 10 โดยเครื่องจะมีเลข 7 ไว้เปรียบเทียบ

ในบรรทัดที่ 40 ถ้าตัวเลขที่เราให้เท่ากับ 7 โปรแกรมจะถูกสั่งให้กระโดดไปที่บรรทัดที่ 70 แล้วแสดงคำว่า "YOU ARE RIGHT"แล้วเลยไปที่บรรทัด 80 เป็นการจบโปรแกรม ถ้าตัวเลขที่เราป้อนเข้าไปไม่ใช่ 7 ซึ่งก็ต้องมากหรือน้อยกว่า 7 เครื่องผ่านบรรทัดที่ 40 ไปถึงบรรทัดที่ 50 ถ้าตัวเลขน้อยกว่า 7 ก็จะติดอยู่ที่บรรทัดที่ 50 แล้วจะถูกสั่งให้กระโดดไปที่บรรทัดที่ 90 ซึ่งจะแสดงออกมาว่าตัวเลขต่ำไป ถ้าตัวเลขมากกว่า 7 จะผ่านการทดสอบของบรรทัดที่ 50 ไป และไปอยู่ที่บรรทัดที่ 60 ซึ่งบรรทัดนี้ จะสั่งให้โปรแกรมกระโดดไปบรรทัดที่ 110 บอกว่าตัวเลขสูงไปซึ่งใต้บรรทัดที่ 90 และ

110 จะมีคำสั่งให้โปรแกรมกลับไปที่บรรทัดที่ 20 คือ เริ่มต้นเดาตัวเลขกันใหม่ เพราะตัวเลขที่เดานั้นสูงไปหรือต่ำไป

คำ GOTO ที่ตามหลังคำ THEN จะเว้นเสียก็ได้ เช่น

```
40   IF   A  =  7   THEN   70
```

12.5 เครื่องหมายเปรียบเทียบที่ใช้กับ IF ... THEN ... มีดังนี้

A = B	หมายความเลขจำนวน A เท่ากับ	เลขจำนวน B
A<> B	หมายความเลขจำนวน A ไม่เท่ากับ	เลขจำนวน B
A < B	หมายความเลขจำนวน A มีค่าน้อยกว่า	เลขจำนวน B
A > B	หมายความเลขจำนวน A มีค่ามากกว่า	เลขจำนวน B
A <= B	หมายความเลขจำนวน A น้อยกว่าหรือเท่ากับ	เลขจำนวน B
A >= B	หมายความเลขจำนวน A มากกว่าหรือเท่ากับ	เลขจำนวน B

12.6 พิมพ์โปรแกรมข้างล่างนี้

```
10  REM     PROGRAM #12-2
20 A = 0
30 A = A + 1
40  PRINT A
50  IF A < 10 THEN  GOTO 30
60  PRINT "END"
70  END


]RUN
1
2
3
4
5
6
```

บรรทัดที่ 20 เรากำหนดให้ตัวแปร A มีค่าเท่ากับ 0

บรรทัดต่อมา เรากำหนดให้นำค่าในตัวแปร A มาบวกเข้ากับ 1 แล้วเอาค่าที่ได้เข้าไปแทนค่าในตัวแปร A เดิม

บรรทัดที่ 40 ให้แสดงค่าของ A บนจอภาพ

```
7
8
9
10
END
```

บรรทัดที่ 50 เครื่องจะนำค่าในตัวแปร A ไปเปรียบเทียบว่ามาก หรือน้อยกว่า 10 ถ้าน้อยกว่า 10 ให้ย้อนกลับไปบรรทัดที่ 30 ใหม่ ค่าในตัวแปร A จะเปลี่ยนจาก 0 เป็น 1 และ 2 ตามจำนวนรอบ เมื่อค่าในตัวแปร A เป็น 10 โปรแกรมจะไม่กลับไปที่บรรทัดที่ 30 จะผ่านไปที่บรรทัด 60 แล้ว แสดงคำว่า END บนจอแล้วหยุดที่บรรทัด 70

ตารางข้างล่างนี้ แสดงการทำงานของเครื่องแต่ละรอบ

	รอบแรก	รอบที่ 2	รอบที่ 3	รอบที่ 10
บรรทัดที่ 20	A = 0			
บรรทัดที่ 30	A = 0+1	A = 1+1	A = 2+1	A = 9+1
บรรทัดที่ 40	แสดงค่า A=1	แสดงค่า A=2	แสดงค่า A=3	แสดงค่า A=10
บรรทัดที่ 50	1 น้อยกว่า 10 กลับไปบรรทัดที่ 30	2 น้อยกว่า 10 กลับไปบรรทัดที่ 30	3 น้อยกว่า 10 กลับไปบรรทัดที่ 30	10 เท่ากับ 10 เลยมาบรรทัดที่ 60
บรรทัดที่ 60				แสดงคำ END
บรรทัดที่ 70				จบโปรแกรม

รูปที่ 12.1 วงรอบ

12.7 ตัวอย่างข้างล่างนี้ เป็นตัวอย่างการใช้ IF.... THEN....

```
50   IF   A  =  10   THEN  END
50   IF   A <> C + D  THEN PRINT "WIN"
50   IF   A  =  8   THEN   B  =  0
```

สรุป คำสั่ง IF.... THEN.... ใช้กับเครื่องหมายเปรียบเทียบ ซึ่งตามมาข้างหลัง IF ถ้าการเปรียบเทียบเป็นจริงจึงจะปฏิบัติตามคำสั่งหลัง THEN หากไม่เป็นจริงจะผ่านไปบรรทัดถัดมา

แบบฝึกหัดบทที่ 12

12.1 จงเขียนโปรแกรมการจ่ายเงินของฉลองและสมควร ตามข้อ 12.1 โดยใช้ IF.... THEN....

เฉลยแบบฝึกหัดบทที่ 12

12.1

```
5  REM  S12.1
10  HOME
20  INPUT "HOW MUCH MONEY HAS CHA
     LONG GOT ?  ";A
30  INPUT "HOW MUCH SOMKUAN GOT ?
       ";B
40  PRINT
50  IF A > B THEN  PRINT "CHALONG
      PAYS"
60  IF A < B THEN  PRINT "SOMKUAN
      PAYS"
70  IF A = B THEN  PRINT "AMERICA
     N SHARE"

]RUN
HOW MUCH MONEY HAS CHALONG GOT ?  350
HOW MUCH SOMKUAN GOT ?  300

CHALONG PAYS

]RUN
HOW MUCH MONEY HAS CHALONG GOT ?  350
HOW MUCH SOMKUAN GOT ?  1350

SOMKUAN PAYS

]RUN
HOW MUCH MONEY HAS CHALONG GOT ?  350
HOW MUCH SOMKUAN GOT ?  350

AMERICAN SHARE
```

13

สลับฉาก

13.1 เราได้เคร่งเครียดกับการเขียนโปรแกรมมาพอสมควรแล้ว ลองหาอะไรมาตกแต่งโปรแกรมให้สวยหรู น่าดูบ้างจะได้ไม่เบื่อ

13.2 SPEED = ...

ถ้าเราเอา SPEED = ตามหลังด้วยตัวเลขตั้งแต่ 1 ถึง 255 เข้าเป็นบรรทัดหนึ่งของโปรแกรม เราจะได้เห็นการทำงานของโปรแกรมที่ช้าลง ตัวเลขน้อยจะทำงานช้า ตัวเลขมากจะทำงานเร็ว และตัวเลขที่ทำงานตามปกติของเครื่องคือ 255

```
5  HOME
10  REM   PROGRAM#13-1
20  SPEED= 50
30  PRINT "DEMONSTRATION FOR SPEE
     D"
40  SPEED= 255
50  END

]RUN
DEMONSTRATION FOR SPEED
```

บรรทัดที่ 20 เรากำหนดให้ SPEED = 50

บรรทัดที่ 30 ลองเปลี่ยนค่าในเครื่องหมายคำพูดในบรรทัดที่ 30 เป็นชื่อคุณเอง แล้วดูชื่อคุณค่อยๆ ปรากฏมา บรรทัดที่ 40 เปลี่ยนให้ SPEED เป็นไปตามอัตราปกติ มิฉะนั้นการกระทำใดๆ ของคุณหลังจากบรรทัดที่ 20 จะช้า

ไปหมด นอกจากจะปิดเครื่อง ลองเปลี่ยนบรรทัดที่ 20 ใหม่ แล้วเพิ่มไปอีก 1 บรรทัด เป็นบรรทัดที่ 25 แล้วเปลี่ยนชื่อในบรรทัดที่ 30 เป็นชื่อแฟนคุณ แล้วลอง RUN ดู เครื่องจะคอยข้อมูลที่บรรทัดที่ 20 คุณลองป้อนตัวเลขระหว่าง 0 ถึง 255 แล้วคอยสังเกตความเร็วดู

```
20 INPUT"SPEED=  ";A

25 SPEED=A

50 GOTO 20
```

```
10  REM   PROGRAM #13-2
20  INPUT "SPEED= ";A
25  SPEED= A
30  PRINT "DEMONSTRATION FOR SPEE
     D"
40  SPEED= 255
50  GOTO 20

]RUN
SPEED= 10
DEMONSTRATION FOR SPEED
SPEED= 200
DEMONSTRATION FOR SPEED
```

13.3 อย่าเพิ่งลบโปรแกรม # 13-2 กด RESET แล้วเพิ่มบรรทัดที่ 26 และ 35

```
26 INVERSE

35 NORMAL
```

ลอง RUN ดู คุณจะเห็นภาพที่แปลกตา

13.4 กด RESET แล้วเปลี่ยนบรรทัดที่ 26 เป็น FLASH

```
26 FLASH
```

ลอง RUN ดู คุณจะได้ภาพอีกแบบหนึ่ง คำ NORMAL ในบรรทัดที่ 35 เป็นสิ่งจำ

เป็น มิฉะนั้นการทำงานของเครื่องจะไม่อยู่ในสภาพเดิม

13.5 พิมพ์โปรแกรมบรรทัดนี้ลงไป

```
NEW
10 PRINT CHR$ (7)
RUN
```

ลอง RUN ดู คุณจะได้เสียงดังปิ๊บขึ้น โปรแกรมของคุณจะสดใสซาบซ่าขึ้น ถ้าคุณเลือกใส่บรรทัดนี้ให้ถูกต้องตามกาละ

14

FOR.... NEXT....

14.1 สังเกตโปรแกรมนี้

```
 10  REM  PRO#14-1
 20 A = 2
→30  PRINT A
 40 A = A + 2
 50  IF A <  = 10 THEN  GOTO 30

 ]RUN
 2
 4
 6
 8
 10
```

โปรแกรมนี้ เริ่มด้วยกำหนดให้ตัวแปร A มีค่าเท่ากับ 2 บรรทัดที่ 40 ให้เอา 2 บวกเข้ากับค่าในตัวแปร A ได้เท่าใดให้เอาเข้าไปไว้ในตัวแปร A บรรทัดที่ 50 จะสั่งให้เอาค่าใน A ไปตรวจสอบดูว่า หากค่าใน A น้อยกว่าหรือเท่ากับ 10 ให้โปรแกรมกลับไปที่บรรทัดที่ 30 คือพิมพ์ค่าตัวเลขในตัวแปร A ถ้าค่าใน A มากกว่า 10 โปรแกรมจะผ่านไปและจบโปรแกรม

ในโปรแกรมนี้ โปรแกรมจะหมุนเป็นวงรอบอยู่ระหว่างบรรทัดที่ 30 และ 50 ค่าในตัวแปร A จะมีค่า 2 ในรอบแรก เพิ่มด้วยการบวก 2 เป็น 4 ในรอบที่ 2 และเรื่อยไปจนถึง 10 จึงออกจากวงรอบ

14.2 โปรแกรม 14.1 เราอาจจะเขียนใหม่ได้ดังนี้

```
10   REM   PRO#14-2
20   FOR A = 2 TO 10 STEP 2
30   PRINT A
40   NEXT A


]RUN
2
4
6
8
10
```

โปรแกรมทั้งสองนี้ ให้ผลเหมือนกันทุกประการ แต่การเขียนโปรแกรม จะสั้นกว่า

สังเกตดูรูปแบบของบรรทัดที่ 20 และ 40

```
→ 20   FOR A = I TO E STEP S
  30
  40   NEXT A
```

วงรอบนี้เรียกว่า วงรอบอัตโนมัติ

เรามาดูความหมายของแต่ละคำ

FOR เป็นคำที่บ่งถึงการเริ่มต้นวงรอบ

A เป็นตัวแปร เรียกว่า ตัวแปรควบคุม

I มีค่าเป็นตัวเลข เรียกว่า ค่าเริ่มต้น

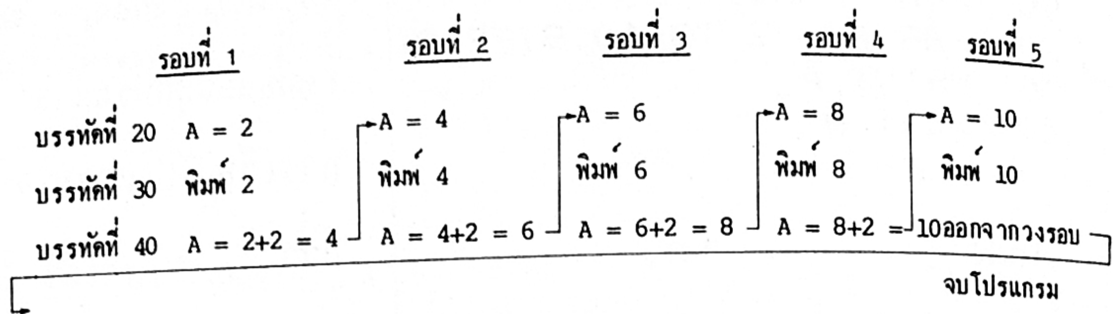
E มีค่าเป็นตัวเลข เรียกว่า ค่าสุดท้าย

S มีค่าเป็นตัวเลข เรียกว่า ค่าเพิ่ม

NEXT A คือส่วนล่างของวงรอบซึ่งจะสั่งให้โปรแกรมย้อนกลับไปยังบรรทัดที่มีคำว่า FOR อยู่ แต่ละรอบค่าของ A จะเพิ่มขึ้นรอบละ +S และจะหมุนอยู่ดังนี้ จนกระทั่งค่าของ A เท่ากับ E จึงจะหลุดจากวงรอบ

ย้อนกลับไปดูบรรทัดที่ 20 ถึง 40 ในโปรแกรมข้อ 14.2

มาดูแผนผังวงรอบอัตโนมัติ ว่าแต่ละรอบจะเกิดอะไรขึ้นบ้าง



รูปที่ 14.1 วงรอบ FOR......NEXT

ตามโปรแกรมนี้ แต่ละรอบค่าของ A จะเพิ่มทีละ 2 (STEP 2) จาก 2 เป็น 4 จนกระทั่งถึง 10 เมื่อถึง 10 ก็จะผ่านบรรทัดที่ 40 ไปและจบโปรแกรมเลย

14.3 สำหรับ STEP ที่มีค่าเป็น 1 อาจจะเว้นไม่เขียน STEP 1 ก็ได้ เช่น

```
60 FOR I=1 TO 10


100 NEXT I
```

14.4 ในกรณีที่เป็นวงรอบเดียว เราอาจใช้คำว่า NEXT เฉยๆ โดยไม่ต้องมีตัวแปรตามก็ได้ เช่น

```
30 FOR I=2 TO 8 STEP 2


60 NEXT
```

14.5 ค่าเพิ่ม (STEP) อาจจะมีค่าเป็นลบหรือเป็นทศนิยมก็ได้ เช่น

```
50 FOR I=12.5 TO 2.5 STEP -2.5



80 NEXT
```

14.6 วงรอบอาจจะซ้อนกันอยู่ก็ได้ ซึ่งเรียกว่า วงรอบซ้อน (NESTED LOOP)

```
10   REM   PRO#14-3
20   PRINT "MULTIPLICATION TABLE"
30   PRINT
40   FOR I = 1 TO 10
50   FOR J = 1 TO 3
60   PRINT (I * J),
70   NEXT J
80   NEXT I

]RUN
MULTIPLICATION TABLE

1                 2                 3
2                 4                 6
3                 6                 9
4                 8                 12
5                 10                15
6                 12                18
7                 14                21
8                 16                24
9                 18                27
10                20                30
```

ในโปรแกรมนี้ บรรทัดที่ 50 ถึง 70 เป็นวงรอบหนึ่ง ซึ่งอยู่ภายในวงรอบของบรรทัด 40 ถึง 80

มีข้อความระวังอยู่ที่บรรทัดที่ 70 ระวังอย่าใส่ค่าของตัวแปรควบคุม(J และ I) สลับกัน

เปลี่ยนคำในเครื่องหมายคำพูดเป็นชื่อของท่านเอง แล้ว RUN โปรแกรมนี้ดู

```
10   REM   PRO#14-4
20   FOR I = 1 TO 20
30   PRINT "NAVARAT"
40   NEXT I

]RUN
NAVARAT
NAVARAT
NAVARAT
NAVARAT
NAVARAT
NAVARAT
```

14.7 โปรแกรมถ่วงเวลา

```
5   REM   PRO#14-5
10   HOME
20   SPEED= 1
30   PRINT "SOMBAT"
40   SPEED= 255
50   FOR I = 1 TO 1000
60   NEXT I
70   HOME
```

เปลี่ยนบรรทัดที่ 30 เป็นชื่อของท่านเอง ซึ่งจะปรากฏบนจอทีละตัวเมื่อครบตัวจะอยู่บนจอพักหนึ่ง แล้วจะหายไป โปรแกรมที่ถ่วงเวลา คือ บรรทัดที่ 50 และ 60 เมื่อมาถึงวงรอบนี้ เครื่องจะเสียเวลาวนอยู่จาก 1 ถึง 1,000 รอบ เมื่อครบก็จะหลุดมาบรรทัดที่ 70 ซึ่งสั่งให้หายไป

สรุป คำสั่ง FOR... NEXT... คือ คำสั่งที่ให้โปรแกรมเดินเป็นวงรอบกลับไปกลับมา แล้วหลุดออกจากวงรอบเองโดยอัตโนมัติ

แบบฝึกหัดบทที่ 14

14.1 จงเขียนตารางแปลงค่าของแกลลอนเป็นลิตร ตั้งแต่ 1 แกลลอน จนถึง 20 แกลลอน โดยให้ 1 แกลลอน มีค่าเท่ากับ 3.785 ลิตร

14.2 จงทำตารางแปลงค่าของอุณหภูมิ จากองศาเซลเซียส เป็นองศาฟาเรนไฮท์ ตั้งแต่ 30 องศาเซลเซียส จนถึง 50 องศาเซลเซียส โดยเพิ่มขึ้นทีละ 2 องศา

เฉลยแบบฝึกหัดบทที่ 14

14.1

```
5  REM  S14.1
10  HOME
20 G = 3.785
30  PRINT "GALLON","LITER"
40  PRINT "------","-----"
50  FOR I = 1 TO 20
60  PRINT I,G * I
70  NEXT I

]RUN
GALLON          LITER
------          -----
1               3.785
2               7.57
3               11.355
4               15.14
5               18.925
6               22.71
7               26.495
8               30.28
9               34.065
10              37.85
11              41.635
12              45.42
13              49.205
14              52.99
15              56.775
16              60.56
17              64.345
18              68.13
19              71.915
20              75.7
```

```
5  REM  S14.2
10  HOME : SPEED= 50: FLASH : PRINT
    "CONVERSION TABLE": NORMAL :
     SPEED= 225: PRINT : PRINT "
    DEG.C","DEG.F": PRINT "-----
    ","-----": PRINT
20  FOR I = 30 TO 50 STEP 2
30  PRINT I,(I * 9 / 5) + 32
40  NEXT I

]RUN
CONVERSION TABLE

DEG.C                DEG.F
-----                -----

30                   86
32                   89.6
34                   93.2
36                   96.8
38                   100.4
40                   104
42                   107.6
44                   111.2
46                   114.8
48                   118.4
50                   122
```

ตัวแปรพันธะ: (STRING VARIABLES) 15

15.1 A = 158

ข้อความนี้ไม่ใช่สมการแต่มีความหมายว่าให้นำตัวเลข 158 เข้าไปบรรจุอยู่ในตัวแปรที่มีชื่อว่า A

หากว่าเราจะนำตัวอักษรจำนวนหนึ่งบรรจุเข้าไปในตัวแปรบ้าง เราก็ทำได้โดยเขียนดังนี้

A $ = "SOMBAT AND NAVARAT"

จงสังเกตว่า อักษรกลุ่มนั้นจะต้องอยู่ในเครื่องหมายคำพูด และตัวแปร A จะมีเครื่องหมายดอลลาร์ ($) ตาม ตัวแปร A ซึ่งเรียกว่า VARIABLE จะเปลี่ยนชื่อเป็น STRING VARIABLE ซึ่งต่อไปจะเรียกว่า "ตัวแปรพันธะ" ลักษณะและข้อกำหนดของชื่อตัวแปรพันธะจะเหมือนกับลักษณะของตัวแปร ดังกล่าวไว้ในบทที่ 4 ทุกประการ เว้นแต่จะต้องมี $ ตาม

ตัวอักษรที่อยู่ในเครื่องหมายคำพูดสามารถจะมีได้ตั้งแต่ 0 จนถึง 255 ตัว

เราสามารถเอาตัวเลขทำเป็นตัวแปรพันธะโดยใส่ไว้ในเครื่องหมายคำพูด-ได้ แต่ตัวเลขนั้นจะหมดสภาพการมีค่าไปทันที

A $ = "158"

สิ่งที่อยู่ในตัวแปรพันธะที่มีชื่อ A นั้นไม่ใช่หนึ่งร้อยห้าสิบแปด แต่เป็นเลขหนึ่ง เลขห้า และเลขแปด ซึ่งจะนำไปใช้ในการคำนวณไม่ได้

```
10  REM  PRO#15-1
20 A$ = "SOMBAT"
30  PRINT A$

]RUN
SOMBAT
```

บรรทัดที่ 20 เราสั่งให้เอาคำว่า SOMBAT ไปบรรจุไว้ในตัวแปรพันธะที่มีชื่อ A

บรรทัดที่ 30 สั่งให้แสดงสิ่งที่อยู่ในตัวแปรพันธะชื่อ A นั้น

15.2 เราสามารถจะเอาตัวแปรพันธะมารวมกันได้ โดยการบวกกัน

```
5  REM  PRO#15-2
10  HOME
20 A$ = "THE ACTOR"
30 B$ = "OF THIS FILM"
40 C$ = "IS SOMBAT"
50 D$ = A$ + B$ + C$
60  PRINT D$

]RUN
THE ACTOROF THIS FILMIS SOMBAT
```

บรรทัดที่ 50 เราให้เอาสิ่งที่อยู่ในตัวแปรพันธะ A, B และ C มารวมกันแล้วมาบรรจุใน D เมื่อสั่งให้พิมพ์ออกมา จะปรากฏผลดังแสดงซึ่งจะเห็นว่าวรรคตอนผิดไปหมด

วิธีแก้มีอยู่ 2 วิธีคือ วิธีแรก คือใส่ตัวแปรพันธะที่มีอักษร 0 ตัว (เรียกว่า NULL STRING) "   " (ระหว่างเครื่องหมายคำพูดให้เคาะคาน SPACE BAR หนึ่งครั้ง) ไว้ระหว่างเครื่องหมาย + ในบรรทัดที่ 50 ดังนี้

```
50 D$ = A$ + " " + B$ + " " + C$
```

ลอง RUN ดู

อีกวิธีหนึ่ง ในบรรทัดที่ 20 และ 30 ให้เคาะ SPACE BAR หนึ่งครั้ง ก่อนปิดเครื่องหมายคำพูด

```
20 A$="THE ACTOR "

30 B$="OF THIS FILM "
```

15.3 พิมพ์โปรแกรมนี้

```
5  HOME
10  REM  PRO#15-3
20  INPUT "WHAT IS YOUR NAME? ";A
     $
30  PRINT
40  PRINT A$;" ! WHAT A LOVELY NA
     ME."
50  PRINT
60  INPUT "WHAT ARE YOU STUDYING
     ? ";B$
70  PRINT
80  PRINT B$;" !,QUITE AN INTERES
     TING SUBJECT, GOOD LUCK ";A$
     ;"."

]RUN
WHAT IS YOUR NAME? PRAPAT

PRAPAT ! WHAT A LOVELY NAME.

WHAT ARE YOU STUDYING ? ENGINEERING

ENGINEERING !,QUITE AN INTERESTING SUBJECT,

    GOOD LUCK PRAPAT.
```

โปรแกรมนี้เป็นการโต้ตอบกับเครื่อง โดยเครื่องจะเก็บข้อมูลที่เราตอบไว้ในตัวแปรพันธะ แล้วเครื่องจะเอาตัวแปรพันธะนี้มาใช้ โปรแกรมแบบนี้ พึงระวังวรรคตอน ซึ่งอาจจะแก้ไขได้ตามวิธีที่กล่าวไว้ใน 15.2

15.4 เราสามารถจะนับจำนวนคำในตัวแปรพันธะได้ โดยใช้คำสั่ง LEN ซึ่งย่อมาจาก LENGTH ดังตัวอย่าง

```
5  REM  PRO#15-4
10  HOME
20  INPUT "YOUR NAME ? ";N$
30 A =  LEN (N$)
40  PRINT "YOUR NAME HAS ";A;" LE
     TTERS"
50  FOR I = 1 TO 3000: NEXT I
60  GOTO 10

]RUN
YOUR NAME ? CHALONG
YOUR NAME HAS 7 LETTERS
YOUR NAME ? PRAPAT UTHAYOPAS
YOUR NAME HAS 16 LETTERS
```

บรรทัดที่ 30 คือ การนับจำนวนตัวอักษรในตัวแปรพันธะ

15.5 ลองดูโปรแกรมนี้

```
5  REM  PRO#15-5
10  HOME
20 A$ = "PRAPAT AND DR.PANPIS"
30  PRINT A$: PRINT
40  PRINT "LEFT$(A$,6)", LEFT$ (A
     $,6)
50  PRINT "RIGHT$(A$,9)", RIGHT$
     (A$,9)
60  PRINT "MID$(A$,8,3)", MID$ (A
     $,8,3)

]RUN
PRAPAT AND DR.PANPIS

LEFT$(A$,6)       PRAPAT
RIGHT$(A$,9)      DR.PANPIS
MID$(A$,8,3)      AND
```

สังเกตบรรทัดที่ 40 , 50 และ 60

บรรทัดที่ 40 เป็นการสั่งให้พิมพ์สิ่งที่อยู่ในตัวแปรพันธะ A จากซ้ายมือสุดนับมาทางขวา 6 ตัว

บรรทัดที่ 50 เป็นการสั่งให้พิมพ์ตัวแปรพันธะ A โดยนับจากทางขวาสุดย้อนมาทางซ้าย 9 ตัว และพิมพ์จากซ้ายไปขวาตามลำดับ

บรรทัดที่ 60 เป็นการสั่งให้พิมพ์ตัวแปรพันธะ A โดยพิมพ์จากตัวที่ 8 นับจากทางซ้ายแล้วพิมพ์ 3 ตัว

สังเกตดูตามแผนภูมินี้แล้วจะเข้าใจได้ชัด

A$ =

P	R	A	P	A	T		A	N	D		D	R	.	P	A	N	P	I	S
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
											9	8	7	6	5	4	3	2	1

LEFT$(A$,6)

RIGHT$(A$,9)

MID$(A$,8,3)

สรุป 1. ตัวแปรพันธะคือ ตัวอักษรหรือตัวเลขที่อยู่ในเครื่องหมายคำพูด ที่เราสามารถจะนำไปบรรจุไว้ในหน่วยความจำ แล้วให้ชื่อไว้ เราสามารถจะเรียกตัวแปรกลุ่มนี้มาใช้เมื่อใดก็ได้ โดยระบุชื่อของตัวแปรพันธะนั้น

2. เราสามารถจะนับจำนวนของตัวอักษร หรือตัวเลขของตัวแปรพันธะนี้ได้ โดยใช้คำสั่ง LEN

3. เราสามารถจะให้เครื่องแสดงส่วนใดส่วนหนึ่งของตัวแปรพันธะได้ โดยใช้คำสั่ง LEFT, RIGHT หรือ MID

แบบฝึกหัดบทที่ 15

15.1 จงเขียนโปรแกรมให้ผู้ใช้เครื่องบอกชื่อ อายุ เพศ ตำบลบ้าน แล้วให้เครื่องพูดคุยกับผู้ใช้

เฉลยแบบฝึกหัดบทที่ 15

15.1

```
5  REM  S15.1
10  HOME
20  INPUT "YOUR NAME ? ";N$
30  INPUT "AGE ?  ";A
40  INPUT "SEX ? M OR F ";S$
50  INPUT "WHERE ABOUT DO YOU LIV
     E ?  ";H$
60  IF S$ = "M" THEN 90
70  IF S$ = "F" THEN 110
80  GOTO 40
90  IF A <  = 18 THEN C$ = "BOY":
     GOTO 130
100 C$ = "GENTLEMAN": GOTO 130
110  IF A <  = 18 THEN C$ = "GIRL
     ": GOTO 130
120 C$ = "LADY"
130  PRINT : SPEED= 80
140  PRINT "HELLO  ";N$
150  PRINT "YOU ARE A GOOD LOOKIN
     G  ";C$
160  PRINT "I LIKE YOUR HOME TOWN
     "
170  PRINT H$;" IS A NICE PLACE"
180  SPEED= 255
190  FOR I = 1 TO 3000: NEXT
200  GOTO 10

]RUN
YOUR NAME ? SOMBAT MEETANAE
AGE ?  38
SEX ? M OR F M
WHERE ABOUT DO YOU LIVE ?  BANGKOKNOI

HELLO  SOMBAT MEETANAE
YOU ARE A GOOD LOOKING  GENTLEMAN
I LIKE YOUR HOME TOWN
BANGKOKNOI IS A NICE PLACE
```

```
YOUR NAME ? NAVARAT SAE TEOW
AGE ?  16
SEX ? M OR F F
WHERE ABOUT DO YOU LIVE ?  BANGLUMPOO

HELLO  NAVARAT SAE TEOW
YOU ARE A GOOD LOOKING  GIRL
I LIKE YOUR HOME TOWN
BANGLUMPOO IS A NICE PLACE
```

SUBROUTINES (โปรแกรมย่อย) 16

16.1 สังเกตโปรแกรมนี้

```
5  REM PRO#16-1
10  HOME
20  PRINT "AREA OF RECTANGLE": PRINT

30  INPUT "LENGTH IN FEET ";L
40 L = L * .3048: REM  CONVERSE T
     O METER
45  PRINT "LENGTH IN METER= ";L: PRINT

50  INPUT "WIDTH IN FEET ";W
60 W = W * .3048: REM  CONVERSE T
     O METER
65  PRINT "WIDTH IN METER= ";W: PRINT

70  PRINT "AREA IN SQ.M.= ";L * W

]RUN
AREA OF RECTANGLE

LENGTH IN FEET 123
LENGTH IN METER= 37.4904

WIDTH IN FEET 65
WIDTH IN METER= 19.812

AREA IN SQ.M.= 742.759804
```

บรรทัดที่ 40 และ 60 เป็นการเปลี่ยนความยาวและความกว้างจากฟุตเป็นเมตร โดยการคูณด้วย .3048 จะสังเกตเห็นว่า 2 บรรทัดนี้ทำงานแบบเดียวกัน

เพื่อหลีกเลี่ยงการเขียนซ้ำซาก เราอาจจะเขียนเป็นโปรแกรมย่อยเพียงกลุ่มเดียว เราสามารถเอามาใช้หลายๆ ครั้งได้ โปรแกรมย่อยนี้เรียกว่า SUBROUTINE

16.2 ดูโปรแกรมที่ 16-2

```
5  REM  PRO#16-2
6  HOME
10  PRINT "SUBROUTINE DEMO": PRINT

20  INPUT "LENGTH IN FEET ";A
30  GOSUB 200
40 L = A
50  INPUT "WIDTH IN FEET ";A
60  GOSUB 200
70 W = A
80  INPUT "HEIGHT IN FEET ";A
90  GOSUB 200
100 H = A
110  PRINT "VOLUME IN CU.M.= ";L *
     W * H
120  END
200  REM  CONVERSION OF FT. TO M.

210 A = A * .3048
220  RETURN

]RUN
SUBROUTINE DEMO

LENGTH IN FEET 124
WIDTH IN FEET 45
HEIGHT IN FEET 8
VOLUME IN CU.M.= 1264.06403
```

โปรแกรมนี้มีข้อสังเกตดังนี้

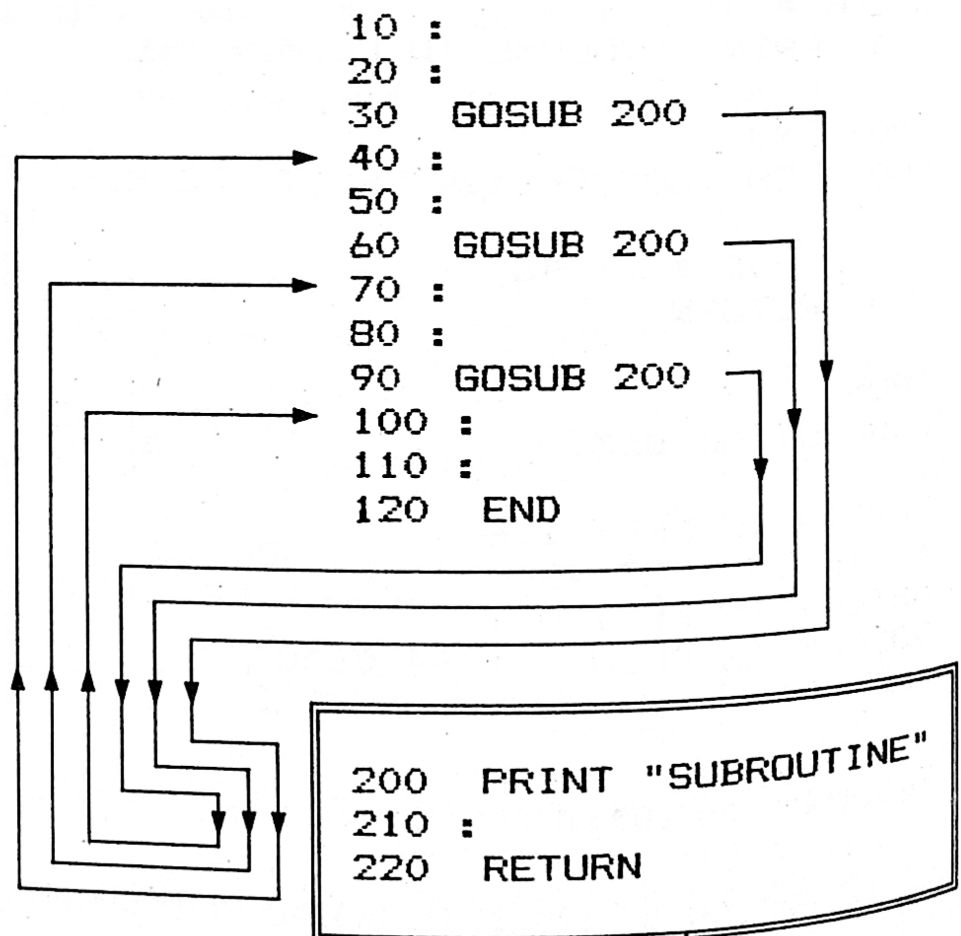
1. บรรทัดที่ 200 ถึง 220 เป็นโปรแกรมย่อย(SUBROUTINE) เป็นโปรแกรมที่ใช้ทำงานในโปรแกรมนี้ถึง 3 ครั้ง โปรแกรมย่อยอาจจะมีตั้งแต่

บรรทัดเดียวจนถึงเป็นร้อยๆ พันๆ บรรทัด ซึ่งการกระทำดังนี้จะประหยัดการเขียนโปรแกรมซ้ำซากเป็นอันมาก และโปรแกรมย่อยอาจจะมีหลายๆ โปรแกรมย่อยก็ได้

2. บรรทัดที่ 30, 60, 90 คำสั่ง GOSUB เป็นคำสั่งที่ให้โปรแกรมกระโดดจากบรรทัดดังกล่าวเข้าไปในโปรแกรมย่อยคือ บรรทัดที่ 400 และเมื่อโปรแกรมผ่านไปจนถึงคำสั่ง RETURN โปรแกรมจะกระโดดกลับไปที่บรรทัดถัดมาจากคำสั่ง GOSUB (ในกรณีนี้คือ บรรทัดที่ 40, 70 และ 100)

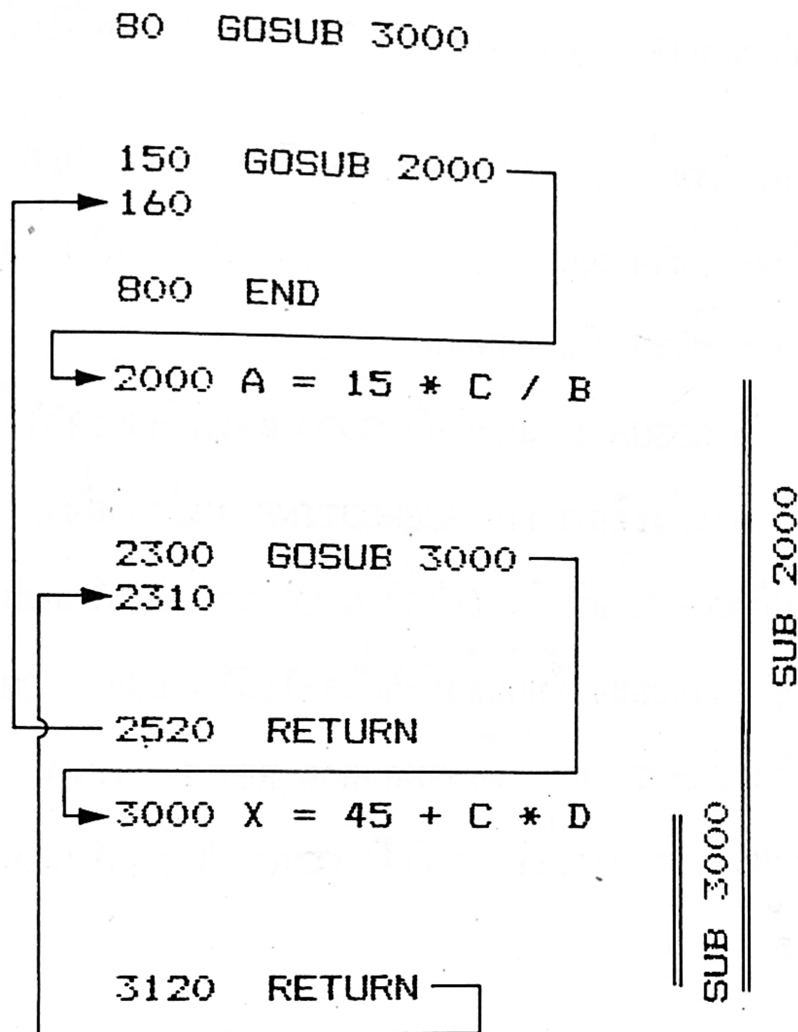
3. จำเป็นอย่างยิ่งที่จะต้องมี END ที่บรรทัดก่อนที่จะถึงโปรแกรมย่อย คือ บรรทัดที่ 120 มิฉะนั้นโปรแกรมจะวิ่งเลยไปถึง 220 แล้วไม่มีที่กลับ

ดูแผนภูมิดังรูปที่ 16.1



รูปที่ 16.1 SUBROUTINE

16.3



รูปที่ 16.2 NESTED SUBROUTINE

ในโปรแกรมย่อย อาจจะโปรแกรมย่อยซ้อนกันอยู่หลายๆ ชั้นก็ได้ จงสังเกตส่วนหนึ่งของโปรแกรมทำนองนี้

บรรทัดที่ 80 มีการสั่งให้ไปที่โปรแกรมย่อย 3000 เครื่องจะข้ามไปที่บรรทัด 3000 และดำเนินไปจนถึง 3120 เมื่อพบคำสั่ง RETURN ก็จะกลับไปที่บรรทัดที่ถัดลงมาจากบรรทัดที่ 80

โปรแกรมย่อย 2000 เป็นโปรแกรมที่ซับซ้อน บรรทัดที่ 150 มีคำสั่งให้ไปโปรแกรมย่อยที่ 2000 เมื่อเครื่องดำเนินไปจนถึงบรรทัดที่ 2300 ก็พบคำ

สั่งให้ไปโปรแกรมย่อยที่ 3000 จึงข้ามไปทำงานที่บรรทัดที่ 3000 จนกระทั่งถึงบรรทัดที่ 3120 พบคำสั่ง RETURN จึงกลับไปเข้าที่บรรทัดที่ 2310 และดำเนินลงมาจนถึงบรรทัดที่ 2520 แล้วจึงย้อนกลับไปที่บรรทัดที่ 160 อีก

สรุป 1. โปรแกรมย่อย (SUBROUTINE) คือโปรแกรมเล็กที่อยู่ในโปรแกรมใหญ่ที่รับหน้าที่ทำงานอย่างเดียวกัน โดยไม่ต้องให้โปรแกรมใหญ่ต้องมีการทำงานซ้ำซากกันหลายตอน

2. คำสั่ง GOSUB เหมือนคำสั่ง GOTO แต่เครื่องจะจำได้ว่ามาจาก GOSUB ใดและเมื่อเดินไปตาม SUBROUTINE จนถึงคำสั่ง RETURN แล้วจะกลับไปที่ GOSUB อันเดิม (คำสั่ง GOTO จะไปแล้วไม่กลับ นอกจากจะสั่ง)

3. โปรแกรมย่อยจะขึ้นต้นด้วยคำสั่งอะไรก็ได้ เช่น REM, FOR.... TO หรือ INPUT แต่จะต้องจบลงด้วย RETURN เสมอ

4. ภายในโปรแกรมย่อยจะมีคำสั่ง GOTO ให้ออกไปนอกโปรแกรมย่อยไม่ได้

แบบฝึกหัดบทที่ 16

16.1 จงเขียนโปรแกรมแปลงค่าระยะจากฟุตเป็นเมตร หรือจากเมตรเป็นฟุตแล้วแต่ผู้ใช้จะเลือก

เฉลยแบบฝึกหัดบทที่ 16

16.1

```
5  REM  S16.1
10  HOME
20  INPUT "DO YOU WANT TO CONVERT
      'FOOT' TO'METER' ? (ANSWER
     YES OR NO)   ";A$
25  PRINT
30  IF A$ = "NO" THEN 110
35  IF A$ <  > "YES" THEN 20
40  PRINT
50  INPUT "O.K. THEN INPUT VALUE
     IN FT. ";F
60  GOSUB 1000
70  PRINT : PRINT
80  INPUT "ANY MORE ? (Y OR N) ";
     B$
85  PRINT
90  IF B$ = "Y" THEN 20
95  HOME
100  END
110  INPUT "O.K. THEN INPUT VALUE
      IN M.";M
120  GOSUB 2000
130  PRINT : PRINT
140  GOTO 80
150 :
1000  REM  CONVERT FT. TO M.
1010 X = F * .3048
1015  INVERSE : PRINT
1020  PRINT F;" FT. IS ";X;" M."
1025  NORMAL
1030  PRINT : PRINT
1040  RETURN
2000  REM  CONVERT M. TO FT.
2010 Y = M * 3.2808
2015  INVERSE : PRINT
2020  PRINT M;" M. IS ";Y;" FT."
2025  NORMAL
2030  PRINT : PRINT
2040  RETURN
```

```
]RUN
DO YOU WANT TO CONVERT 'FOOT' TO'METER' ?
(ANSWER YES OR NO)  YES


 O.K. THEN INPUT VALUE IN FT. 135

 135 FT. IS 41.148 M.



 ANY MORE ? (Y OR N) Y

 DO YOU WANT TO CONVERT 'FOOT' TO'METER' ?

    (ANSWER YES OR NO)  NO

 O.K. THEN INPUT VALUE IN M.42

 42 M. IS 137.7936 FT.
```

ON....GOTO....
ON....GOSUB....

17

17.1 สังเกตโปรแกรมนี้

```
5  REM  PRO#17-1
10  HOME
15 A$ = "*************************
    *****"
20  PRINT "      CALCULATION OF AR
    EA"
25  PRINT "      -----------------
    --"
30  PRINT "1-TRIANGLE"
40  PRINT "2-RECTANGLE"
50  PRINT "3-CIRCLE"
55  PRINT
60  INPUT "TYPE YOUR CHOICE 1,2 O
    R 3    ";X
70  IF X = 1 THEN 200
80  IF X = 2 THEN 300
90  IF X = 3 THEN 400
200  HOME : PRINT A$
210  PRINT "AREA OF TRIANGLE"
220  PRINT A$
230  INPUT "BASE=  ";B
240  INPUT "HEIGHT=  ";H
245  PRINT
250  PRINT "AREA =  ";B * H / 2
260  PRINT : PRINT
270  GOTO 20
300  HOME : PRINT A$
310  PRINT "AREA OF RECTANGLE"
320  PRINT A$
330  INPUT "LENGTH=  ";L
340  INPUT "WIDTH=  ";W
345  PRINT
350  PRINT "AREA =  "L * W
```

```
360  PRINT : PRINT
370  GOTO 20
400  HOME : PRINT A$
410  PRINT "AREA OF CIRCLE"
420  PRINT A$
430  INPUT "RADIUS=  ";R
435  PRINT
440  PRINT "AREA =  ";3.1415 * R ^
     2
450  PRINT : PRINT
460  GOTO 20

]RUN
          CALCULATION OF AREA
          -------------------
1-TRIANGLE
2-RECTANGLE
3-CIRCLE

TYPE YOUR CHOICE 1,2 OR 3    1
*****************************
AREA OF TRIANGLE
*****************************
BASE=  160
HEIGHT=  20

AREA =  1600


          CALCULATION OF AREA
          -------------------
1-TRIANGLE
2-RECTANGLE
3-CIRCLE

TYPE YOUR CHOICE 1,2 OR 3    2
*****************************
AREA OF RECTANGLE
*****************************
LENGTH=  12
WIDTH=  20

AREA =  240
```

```
     CALCULATION OF AREA
     -------------------

1-TRIANGLE
2-RECTANGLE
3-CIRCLE

TYPE YOUR CHOICE 1,2 OR 3     3
*****************************
AREA OF CIRCLE
*****************************
RADIUS=  100

AREA =  31415
```

โปรแกรมนี้เป็นโปรแกรมสำหรับคำนวณพื้นที่ของรูปสามเหลี่ยม สี่เหลี่ยมและวงกลม โดยให้เลือกเอาว่าจะเอาอย่างใด โดยการพิมพ์ 1 หรือ 2 หรือ 3 โปรแกรม บรรทัดที่ 70, 80 และ 90 เป็นการสั่งให้กระโดด โดยมีข้อแม้(CONDITIONAL JUMP) ให้โปรแกรมแยกบรรทัดต่างๆ ตามที่เลือก และหลังจากเลือก และคำนวณพร้อมกับแสดงผลออกมาแล้ว โปรแกรมจะย้อนกลับมาบรรทัดที่ 20 ลอง RUN แล้วแทนค่าดู

17.2 ในโปรแกรมเดิม บรรทัดที่ 70, 80 และ 90 อาจจะลดเหลือบรรทัดเดียวดังนี้

```
70 ON X GOTO 200, 300, 400
80      RETURN
90      RETURN
```

(อย่าลืมกด 80 RETURN และ 90 RETURN เพื่อลบบรรทัดที่ 80 และ 90 ออก) ลอง RUN ออกมาจะเห็นว่าค่าได้คงเดิม

โปรแกรมบรรทัดที่ 70 ทำงานดังนี้

ถ้า X = 1 จะโดดไปบรรทัดที่ 200

ถ้า X = 2 จะโดดไปบรรทัดที่ 300

ถ้า X = 3 จะโดดไปบรรทัดที่ 400

ถ้า X มากกว่า 3 (มากกว่าจำนวนหมายเลขบรรทัดที่ตามหลัง GOTO) โปรแกรมจะผ่านไปบรรทัดต่อไป

ถ้าค่าของ X เป็นทศนิยมเศษจะถูกปัดทิ้ง เครื่องจะเอาจำนวนเต็มมาพิจารณา

17.3 เราสามารถใช้คำสั่งดังนี้กับโปรแกรมย่อย (SUBROUTINE) ได้ดังนี้

285 ON N GOSUB 800, 900, 2 000

เมื่อโปรแกรมผ่านมาถึงบรรทัดนี้

ถ้า N = 1 จะโดดไปที่โปรแกรมย่อย (SUBROUTINE) บรรทัดที่ 800

ถ้า N = 2 จะโดดไปที่โปรแกรมย่อย (SUBROUTINE) บรรทัดที่ 900

ถ้า N = 3 จะโดดไปที่โปรแกรมย่อย (SUBROUTINE) บรรทัดที่ 2 000

ถ้า N = 2.8 จะโดดไปที่โปรแกรมย่อย (SUBROUTINE) บรรทัดที่ 900

ถ้า N = 3.1 จะโดดไปที่โปรแกรมย่อย (SUBROUTINE) บรรทัดที่ 2 000

ถ้า N = 4 หรือมากกว่า โปรแกรมจะผ่านลงมาบรรทัดถัดไป

แบบฝึกหัดบทที่ 17

17.1 จงเขียนโปรแกรม กำหนดให้มีห้อง 3 ห้อง ห้องที่ 1 มีเกมเล่น ห้องที่ 2 มีขนมอร่อย ห้องที่ 3 มีผี ให้ผู้ใช้เครื่องเลือกห้องตามชอบ

เฉลยแบบฝึกหัดบทที่ 17

17.1

```
5  REM  S17.1
10  HOME
20  GOSUB 1000
30  PRINT "          T H R E E    R
      O O M S"
40  GOSUB 1000
50  PRINT "       THERE ARE THREE R
     OOMS": PRINT : PRINT
60  PRINT "       YOU CAN CHOOSE ON
     E": PRINT
70  PRINT " NO. 1   OR   2   OR
      3": PRINT : PRINT
80  PRINT "    TYPE THE NUMBER YOU
      WANT"
90  GET A
100  HOME : PRINT : PRINT : PRINT

200  GOSUB 1000
210  ON A GOSUB 300,400,500
220  GOTO 40
300  PRINT "       ROOM NO. 1": PRINT
     : SPEED= 50
310  PRINT "LOTS OF GAMES TO PLAY
     ": PRINT : PRINT : PRINT "EN
     JOY YOURSELF": SPEED= 255
320  FOR I = 1 TO 2000: NEXT : HOME

330  RETURN
400  PRINT "       ROOM NO.2": PRINT
     : SPEED= 50
410  PRINT "HERE ARE COOKIES FOR
     YOU": SPEED= 255
420  FOR I = 1 TO 2000: NEXT : HOME

430  RETURN
500  SPEED= 10: PRINT "DID YOU CH
     OOSE ROOM NO. 3 ?": PRINT : SPEED= 50
510  FLASH : PRINT "A GHOST IS CO
     MING BRR!": NORMAL : SPEED=
     255
```

```
520  FOR I = 1 TO 2000: NEXT : HOME

530  RETURN
1000  FOR I = 1 TO 35
1010  PRINT "#"; CHR$ (7);
1020  NEXT
1025  PRINT
1030  RETURN

]RUN
###################################
        T H R E E    R O O M S
###################################
     THERE ARE THREE ROOMS


     YOU CAN CHOOSE ONE

 NO. 1    OR    2    OR    3


   TYPE THE NUMBER YOU WANT



###################################
     ROOM NO. 1

LOTS OF GAMES TO PLAY


ENJOY YOURSELF


###################################
     THERE ARE THREE ROOMS


     YOU CAN CHOOSE ONE

 NO. 1    OR    2    OR    3


   TYPE THE NUMBER YOU WANT
```

```
#####################################
     ROOM NO.2

HERE ARE COOKIES FOR YOU


#####################################
     THERE ARE THREE ROOMS


     YOU CAN CHOOSE ONE

 NO. 1    OR    2    OR    3


   TYPE THE NUMBER YOU WANT



#####################################
DID YOU CHOOSE ROOM NO. 3 ?

A GHOST IS COMING BRR
#####################################
     THERE ARE THREE ROOMS


     YOU CAN CHOOSE ONE

 NO. 1    OR    2    OR    3


   TYPE THE NUMBER YOU WANT
```

18

TAB, HTAB, VTAB

18.1 คำสั่ง TAB, HTAB, VTAB คือคำสั่งที่กำหนดให้ตัวอักษรที่เราพิมพ์ลงไปบนแป้นพิมพ์ หรือในโปรแกรมปรากฏอยู่ในตำแหน่งใดตำแหน่งหนึ่งของจอภาพได้

18.2 TAB เป็น PROGRAM COMMAND ที่จะต้องอยู่หลังคำว่า PRINT และมีหมายเลขในวงเล็บกำกับข้างหลัง ดังโปรแกรมข้างล่างนี้

```
5  REM  PRO#18-1
10  PRINT  TAB( 30)"OH"
20  PRINT  TAB( 35)"BYE"

]RUN
                              OH
                                   BYE
```

ตัวเลขในวงเล็บที่ตามหลังคำสั่ง TAB นั้น ถ้ามากไปกว่า 40 ตัวที่แสดงทางจอภาพจะเลื่อนไปอีกบรรทัดหนึ่ง ดังแสดงในแผนภูมิข้างล่าง ตัวเลขนี้จะมีได้จนถึง 255 ซึ่งจะอยู่ประมาณกลางของบรรทัดที่ 7

1 2 3	40
41	80
81	120
121	160

171		200
201		240
241	255	

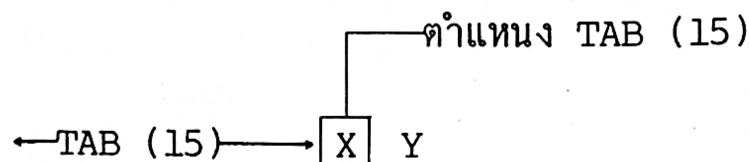
ในบรรทัดเดียวกัน ถ้ามีคำสั่ง TAB หลายคำสั่ง ตัวเลขที่อยู่ในวงเล็บจะต้องเริ่มจากน้อยไปหามาก หรือจากซ้ายไปขวา จะย้อนทางกันไม่ได้ เช่น

```
10 PRINT TAB (10) "X" ; TAB (15) "Y"
```

แต่ถ้าบรรทัดนี้เป็น

```
10 PRINT TAB (15) "X" ; TAB (10) "Y"
```

ตัว Y ที่ปรากฏจะไม่ย้อนต้นไปที่ตำแหน่ง 8 จากซ้าย หากแต่จะมาต่อกับคำที่ TAB (15) ดังนี้



18.3 HTAB เป็นคำสั่งที่คล้ายกับ TAB แต่มีข้อแตกต่างดังนี้

- HTAB จะต้องเขียนในโปรแกรมก่อนบรรทัดที่มีคำสั่ง PRINT เช่น

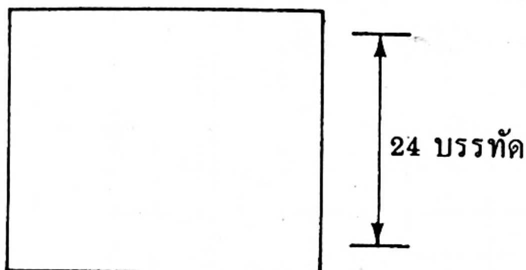
```
10  HTAB  30
20  PRINT  "BYE"
```

- ตัวเลขที่ระบุตำแหน่งของ HTAB ไม่ต้องมีวงเล็บ
- เราสามารถจะสั่งให้แสดงปลายบรรทัดก่อน แล้วย้อนมาแสดงต้นบรรทัดได้ด้วย HTAB ดังเช่น

```
10   REM   PRO#18-2
20   HOME
30   SPEED= 1
40   HTAB 20
50   PRINT "MORNING";
60   HTAB 15
70   PRINT "GOOD"
80   SPEED= 225
```

โปรแกรมนี้สังเกตเครื่องหมาย SEMICOLON ที่บรรทัดที่ 50 สั่งให้ข้อความที่ตามมาจะต้องพิมพ์ลงในบรรทัดเดียวกัน แต่คำสั่งในบรรทัดที่ 60 ให้พิมพ์อยู่ทางต้นบรรทัด

18.4



ทางด้านตั้งของจอภาพ เราสามารถจะเขียนหรือให้แสดงได้ 24 บรรทัด เราสามารถจะสั่งให้แสดงตัวเลข - หรือตัวอักษรที่บรรทัดใดก็ได้ โดยใช้คำสั่ง VTAB

18.5 ลักษณะการเขียน VTAB คล้ายกับ HTAB กล่าวคือ ต้องเขียนในโปรแกรมก่อนบรรทัดที่มีคำสั่ง PRINT ตัวเลขที่ตามหลังซึ่งอยู่ระหว่าง 1 ถึง 24 นั้นไม่ต้องมีวงเล็บ และสามารถจะสั่งให้แสดงบรรทัดใดก็ได้ โดยไม่คำนึงว่าอยู่ส่วนบนหรือส่วนล่างของจอ

```
10   REM   PRO#18-3
20   HOME
30   SPEED= 1
40   VTAB 23
50   PRINT "LINE 23"
60   VTAB 1
70   PRINT "LINE 1"
80   VTAB 12
90   PRINT "LINE 12"
100   SPEED= 255
```

ถ้าตัวเลขหลัง VTAB เกินไปกว่า 24 เครื่องจะบอก "ILLEGAL QUANTITY IN ... "

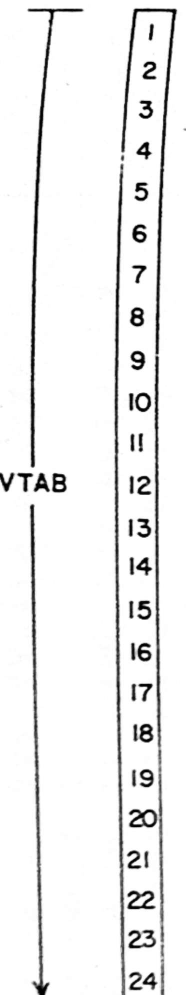
18.6 เราสามารถจะใช้ HTAB และ VTAB ได้ในคราวเดียวกัน

```
10   REM   PRO#18-4
20   REM   SLIDING NAME
30   HOME
40   INPUT "TYPE YOUR NAME  ";N$
50   VTAB 12
60   HTAB 1
70   SPEED= 200
80   FOR I = 1 TO 40 -  LEN (N$)
90 N$ = " " + N$
100  PRINT N$; CHR$ (7);
110  HTAB 1
120  NEXT I
130  SPEED= 255
```

เริ่มแรกโปรแกรมจะให้คุณพิมพ์ชื่อคุณลงไป แล้วเครื่องจะแสดงชื่อคุณบนจอภาพที่บรรทัดที่ 12 แล้วเลื่อนจากทางซ้ายไปขวา ข้อควรสังเกตอยู่ที่บรรทัดที่ 80 ถึง 120

บรรทัดที่ 90 เครื่องจะเพิ่ม NULL STRING (ดู 15.2) ทุกครั้ง ที่วงรอบเคลื่อนไปแต่ละรอบ เมื่อเครื่องแสดงชื่อที่บรรทัดบนจอที่ 12 รอบแรกแล้วจะถูกสั่งให้กลับมาแสดงที่ตัวแรกของบรรทัดที่ 12 อีก แทนที่จะพิมพ์ต่อจากชื่อเดิม (ตามคำสั่งที่มี ; อยู่)

ในการนี้ NULL STRING จะลบตัวอักษรตัวแรกของชื่อออก แล้วพิมพ์ชื่อใหม่ทับตัวเดิม โดยเหลื่อมกัน 1 ตัว(ตัวที่ 1 ของชื่อใหม่ทับตัวที่ 2 ของชื่อเก่า) ดังนี้เรื่อยไปจนถึงช่องที่ 30 จึงหยุด จึงดูเสมือนว่าชื่อวิ่งไป



รูปที่ 18.1 ตำแหน่ง HTAB และ VTAB

โปรแกรมที่ 18.4 นี้มีข้อที่น่าสังเกตอีกในบรรทัดที่ 80 เนื่องจากความกว้างของจอภาพสามารถจะบรรจุตัวอักษรได้ 40 ตัว เรากำหนดให้ชื่อเลื่อนจากตำแหน่ง 1 ไปทางขวา เมื่อสุดจอภาพตัวอักษรตัวสุดท้ายของชื่อควรจะหยุดอยู่ที่ตำแหน่งที่ 40 พอดี หากเกินไปกว่านั้นก็จะวกกลับมาอีกข้างหนึ่ง ทำให้ไม่น่าดู แต่เนื่องจากชื่อคนต่างๆ มีจำนวนตัวอักษรไม่เท่ากัน บรรทัดที่ 80 จึงกำหนดให้พิมพ์ชื่อตั้งแต่ตัวอักษรตัวแรกของชื่อที่พิมพ์ครั้งแรก อยู่ที่ตำแหน่งหนึ่ง และตัวอักษรตัวแรกของชื่อที่พิมพ์ครั้งสุดท้ายอยู่ในตำแหน่ง(40-จำนวนตัวอักษร) ซึ่งก็หมายความว่า ตัวอักษรตัวสุดท้ายของชื่ออยู่ในตำแหน่งที่ 40 พอดี ไม่ว่าชื่อจะยาวหรือสั้นแค่ไหน

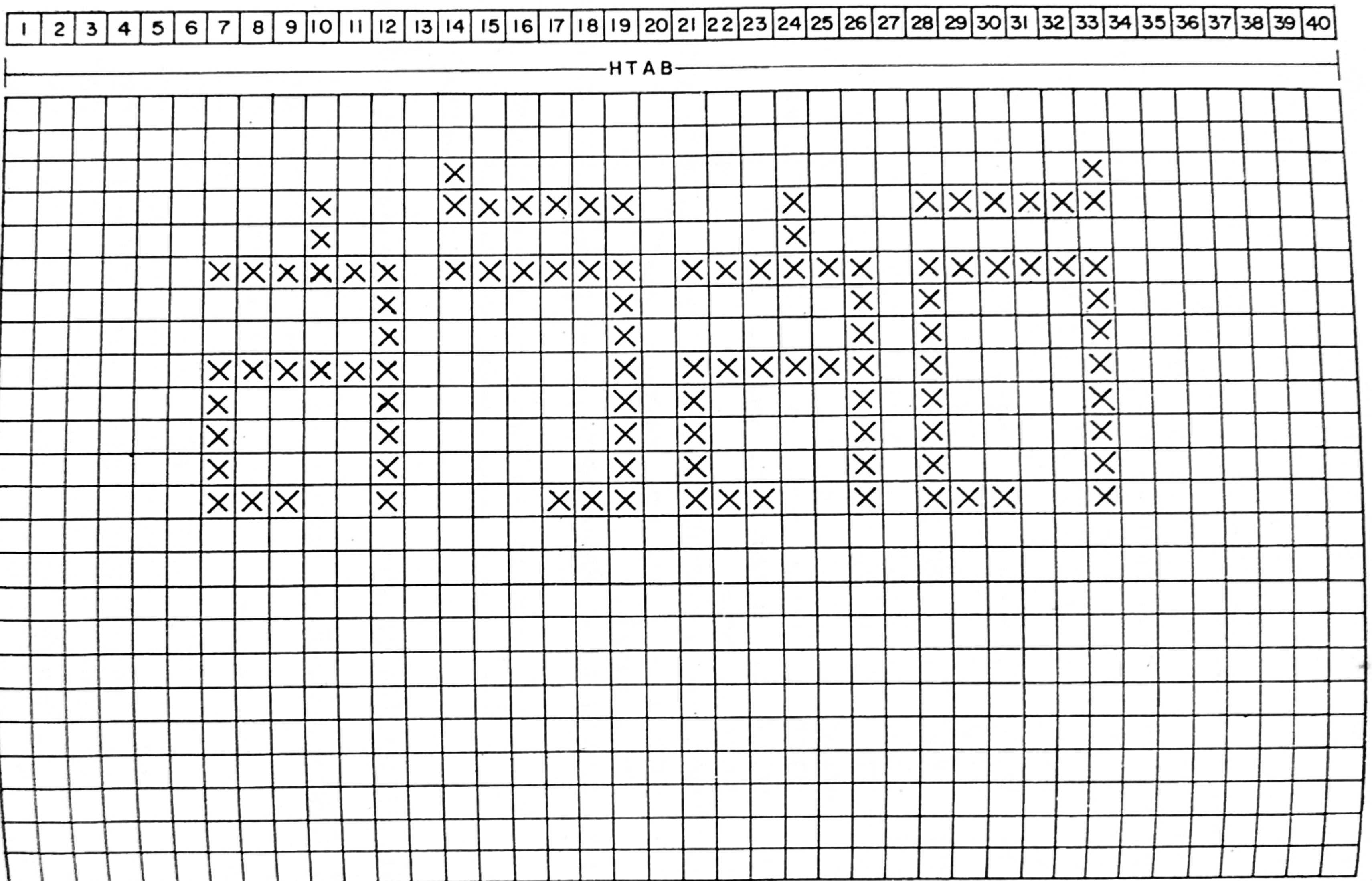
18.7 ตารางที่ 18.1 คือตารางแสดงตำแหน่งของ HTAB และ VTAB

18.8 ลองออกแบบใช้ HTAB และ VTAB พิมพ์คำว่า "สวัสดี" ลงบนจอ ก่อนอื่นต้องออกแบบลงในแผนผังดูก่อน แล้วเขียน PROGRAM ให้พิมพ์ตัวอักษร X ลงบนตำแหน่งต่างๆ

```
10   REM  PRO#18-5
15   HOME : FLASH
16   SPEED= 150
20   FOR I = 6 TO 13
30   VTAB I: HTAB 12: PRINT "X"
40   VTAB I: HTAB 19: PRINT "X"
50   VTAB I: HTAB 26: PRINT "X"
60   VTAB I: HTAB 28: PRINT "X"
70   VTAB I: HTAB 33: PRINT "X"
80   NEXT I
90   FOR I = 9 TO 13
100   VTAB I: HTAB 7: PRINT "X"
110   VTAB I: HTAB 21: PRINT "X"
115   NEXT I
120 Y$ = "XXXXXX":X$ = "XXX"
```

```
130  VTAB 6: HTAB 7: PRINT Y$
131  VTAB 6: HTAB 14: PRINT Y$
132  VTAB 6: HTAB 21: PRINT Y$
133  VTAB 6: HTAB 28: PRINT Y$
140  VTAB 9: HTAB 7: PRINT Y$
150  VTAB 9: HTAB 21: PRINT Y$
160  VTAB 13: HTAB 7: PRINT X$
170  VTAB 13: HTAB 17: PRINT X$
180  VTAB 13: HTAB 21: PRINT X$
190  VTAB 13: HTAB 28: PRINT X$
200  VTAB 4: HTAB 10: PRINT "X"
210  VTAB 5: HTAB 10: PRINT "X"
220  VTAB 4: HTAB 24: PRINT "X"
230  VTAB 5: HTAB 24: PRINT "X"
240  VTAB 3: HTAB 14: PRINT "X"
250  VTAB 4: HTAB 14: PRINT Y$
260  VTAB 4: HTAB 28: PRINT Y$
270  VTAB 3:  HTAB 33: PRINT "X"
280  SPEED= 255
290  NORMAL
300  FOR I = 1 TO 12: PRINT  CHR$
     (7): NEXT
310  FOR I = 1 TO 6000: NEXT : HOME
```

ในโปรแกรมนี้โปรดสังเกตว่า   เราสามารถจะเอาคำสั่งหลายๆ คำสั่งมาเขียนในบรรทัดเดียวกันได้ โดยใช้ : คั่น



รูปที่ 18.2 ตัวอย่างการออกแบบ HTAB, VTAB

แบบฝึกหัดบทที่ 18

18.1 จงเขียนโปรแกรม ให้ผู้ใช้ใส่ชื่อลงไป แล้วชื่อจะเลื่อนจากมุมจอด้านบนซ้ายทแยงมาทางขวามือ จนถึงประมาณกลางจอภาพ แล้วมีชื่อเลื่อนมาจากทางซ้ายมาต่อที่บรรทัดสุดท้าย

เฉลยแบบฝึกหัดบทที่ 18

18.1

```
5  REM  S18.1
10  HOME
20  INPUT "NAME    ";N$
25  HOME
30  SPEED= 200
40  FOR I = 1 TO 14
50  PRINT  TAB( I * 2)N$
60  NEXT I
70  VTAB 15: HTAB 1
80  FOR I = 1 TO 29
90 N$ = " " + N$
100  PRINT N$;
110  HTAB 1
120  NEXT I
130  SPEED= 255
140  PRINT  CHR$ (7)

]RUN
NAME    A
 A
   A
     A
       A
         A
           A
             A
               A
                 A
                   A
                     A
                       A
                         A
                           A
 A A A A A A A A A A A A A A A
```

19

READ.... DATA....

19.1 สังเกตโปรแกรมข้างล่างนี้

```
10  REM  PRO#19-1
20  HOME
30 M$ = "JANUARY EXPENSE"
40 A$ = "1."
50 B$ = "2."
60 C$ = "FOOD"
70 D$ = "GASOLINE"
80 A = 3000
90 B = 850
100 E$ = "BAHT"
110  PRINT  TAB( 10)M$: PRINT
120  PRINT A$;C$,A,E$
130  PRINT B$;D$,B,E$


]RUN
          JANUARY EXPENSE

1.FOOD              3000                BAHT
2.GASOLINE          850                 BAHT
```

โปรแกรมนี้เป็นโปรแกรมที่เขียนอย่างตรงไปตรงมา กำหนดให้ตัวแปรที่บรรจุตัวเลข(NUMERIC VARIABLE) และตัวแปรพันธะ (STRING VARIABLE) มีค่าต่างๆ แล้วสั่งให้พิมพ์ตัวแปรเหล่านั้นเรียงตามลำดับตามต้องการ การเขียนโปรแกรมแบบนี้จะยุ่งยาก

19.2 โปรแกรม 19.1 อาจจะเขียนใหม่ได้ดังนี้

```
10  REM  PRO#19-2
20  HOME
30  DATA  JANUARY EXPENSE,1.,FOOD
     ,3000,BAHT
40  DATA  2.,GASOLINE,850,BAHT
50  READ M$,A$,B$,A,C$,D$,E$,B
60  PRINT  TAB( 10)M$: PRINT
70  PRINT A$;B$,A,C$
80  PRINT D$;E$,B,C$

]RUN
          JANUARY EXPENSE

1.FOOD              3000             BAHT
2.GASOLINE          850              BAHT
```

ในโปรแกรมนี้เราเอาข้อมูลทั้งหมดมาเขียนรวมกันในบรรทัดเดียวกัน โดยมีเครื่องหมาย , คั่น ต่อไปก็สั่งให้มีการอ่านข้อมูลเหล่านั้นเข้าไว้ในตัวแปร แล้วส่งให้พิมพ์ตัวแปรนั้นตามต้องการ คำสั่งที่ทำให้มีการกระทำดังกล่าว คือ

READ... DATA

มีหลักเกณฑ์ดังนี้

1. ข้อมูลที่นำมารวมกันต้องมีคำว่า DATA นำ หลังจากหมายเลขบรรทัด
2. ข้อมูลแต่ละข้อมูลต้องมีเครื่องหมายคอมม่า (,) คั่น นอกจากตัวสุดท้ายต้องไม่มีคอมม่าตาม
3. ข้อมูลที่นำมาเรียงกันจะมีลำดับตั้งแต่ข้อมูลแรก เรียงมาตามลำดับ เมื่อถูกสั่งให้อ่าน เครื่องจะอ่านตั้งแต่ข้อมูลแรก หรือจากซ้ายไปขวา
4. ข้อมูลอาจจะเป็นตัวเลข ตัวอักษรที่อยู่ในเครื่องหมายคำพูดและตัวอักษรที่ไม่มีเครื่องหมายคำพูดก็ได้

5. การบรรจุข้อมูลใช้คำสั่ง READ ตามหลังหมายเลขบรรทัด แล้วตามด้วยตัวแปรที่เป็นประเภทเดียวกับข้อมูล กล่าวคือ ข้อมูลที่เป็นตัวเลขให้อ่านเข้าตัวแปรที่บรรจุตัวเลข(NUMERIC VARIABLE)ข้อมูลที่เป็นกลุ่มอักษรซึ่งรวมทั้งกลุ่มที่อยู่ในเครื่องหมายคำพูด และไม่ได้อยู่ในเครื่องหมายคำพูดก็ตาม ให้อ่านเข้าตัวแปรพันธะ(STRING VARIABLE) ดังนี้

เลข 3,000 อ่านเข้าตัวแปร A

FOOD อ่านเข้าตัวแปร B $

6. จำนวนข้อมูลอาจจะมีมากกว่าตัวแปรที่จะนำมาบรรจุได้ แต่จะน้อยกว่าไม่ได้

7. ข้อมูล(DATA) จะอยู่ส่วนใดของโปรแกรมก็ได้ ไม่จำเป็นจะต้องอยู่ใกล้กับคำสั่ง READ

19.3 ดังที่กล่าวมาในข้อ 3 ของ 19.2 ข้อมูลที่อยู่หลังคำว่า DATA จะจัดเรียงตัวกันไปเป็นลำดับ การอ่านก็เป็นไปตามลำดับ แต่ถ้าจะให้กลับไปอ่านข้อมูลลำดับแรกเรียงลงมาอีก ให้ใช้คำสั่ง RESTORE

```
5  REM  PRO#19-3
10  HOME
20  READ A$,B$,C$,D$
30  READ E$,F$,G$
40  PRINT A$;D$;E$
50  RESTORE
60  READ M$,N$,H$
70  PRINT M$;N$;H$
80  DATA  JACK,AND,JILL,WENT,UP
90  DATA  THE,HILL,30,FEET,HIGH

]RUN
JACKWENTUP
JACKANDJILL
```

โปรแกรมนี้แสดงว่า ตัวแปรที่จะถูกสั่งให้บรรจุ(โดยคำสั่ง READ) อาจจะน้อยกว่าข้อมูล(DATA) ได้

บรรทัดที่ 40 แสดงว่าข้อมูลที่บรรจุในตัวแปรเป็นไปตามลำดับ บรรทัดที่ 60 นั้น เมื่อผ่านคำสั่ง RESTORE (บรรทัดที่ 50) มาแล้ว ก็จะย้อนกลับไปอ่านตัวแรกของ DATA ใหม่

สรุป - คำสั่ง "READ" คือ คำสั่งที่ให้เครื่องอ่าน และเก็บข้อมูลที่มีอยู่ในบรรทัด "DATA"

- ตัวแปรที่เก็บข้อมูลจะต้องเป็นตัวแปรชนิดเดียวกับข้อมูล
- ตัวแปรในคำสั่ง READ จะมากกว่าข้อมูลใน DATA ไม่ได้ แต่ข้อมูลใน DATA จะมากกว่าตัวแปรได้
- ข้อมูลจะอยู่ส่วนใดของโปรแกรมก็ได้
- คำสั่ง RESTORE คือ คำสั่งที่ให้การอ่านข้อมูลย้อนกลับไปที่ข้อมูลตัวแรก

แบบฝึกหัดบทที่ 19

19.1 ในชั้นเรียนแห่งหนึ่งมีนักเรียน 10 คน มีรายชื่อดังต่อไปนี้ สนอง, สมควร, ทองคำ, สุชาติ, ชโลธร, ฉลอง, ชาญชัย, ภุชงค์, สุรศักดิ์, ชูเกียรติ์ และคะแนนสอบของนักเรียนแต่ละคนเรียงตามลำดับมีดังนี้ 58, 46, 52, 64, 83, 70, 65, 90, 78, 86 จงเขียนโปรแกรมเป็นตารางแสดงหมายเลขนักเรียน ชื่อนักเรียน และคะแนน

เฉลยแบบฝึกหัดบทที่ 19

19.1

```
5  REM  S19.1
10  HOME
20  DATA  1,2,3,4,5,6,7,8,9,10
30  DATA  " SANONG"," SOMKUAN","
     THONGKUM"," SUCHART"," CHALO
     THORN"," CHALONG"," CHANCHAI
     "," PUTCHONG"," SURASAK"," C
     HUKIAT"
40  DATA  58,46,52,64,83,70,65,90
     ,78,86
50  READ A,B,C,D,E,F,G,H,I,J
60  READ A$,B$,C$,D$,E$,F$,G$,H$,
     I$,J$
70  READ K,L,M,N,O,P,Q,R,S,T
80  PRINT  TAB( 10)"STUDENT'S MAR
     KS": PRINT
90  PRINT A;A$,K: PRINT B;B$,L: PRINT
     C;C$,M: PRINT D;D$,N: PRINT
     E;E$,O: PRINT F;F$,P: PRINT
     G;G$,Q: PRINT H;H$,R: PRINT
     I;I$,S: PRINT J;J$,T

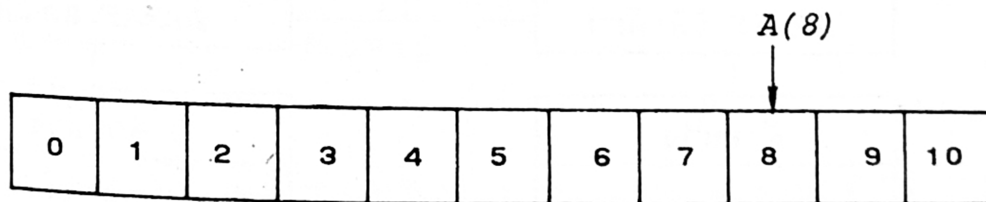
]RUN
          STUDENT'S MARKS

1 SANONG            58
2 SOMKUAN           46
3 THONGKUM          52
4 SUCHART           64
5 CHALOTHORN        83
6 CHALONG           70
7 CHANCHAI          65
8 PUTCHONG          90
9 SURASAK           78
10 CHUKIAT          86
```

20

ARRAY

20.1 นักธุรกิจนายหนึ่งสร้างทาวน์เฮ้าส์ชั้นเดียวขึ้นมา 11 ห้อง เขามีนิสัยรักธรรมชาติ จึงตั้งชื่อห้องต่างๆ เป็นภาษาดอกไม้ ดังนี้ รสสุคนธ์, ชมนาด, บงกช, โกมุท, เฟื่องฟ้า, ฯลฯ ปรากฏว่าเกิดความยุ่งยากขึ้นเพราะไม่สามารถจะจำได้ว่าห้องต่างๆ นี้อยู่ส่วนไหนของแถว และถ้าเกิดจะสร้างขึ้นมาอีกหลายแถวก็จะหาชื่อดอกไม้มาตั้งไม่ได้ เขาจึงเปลี่ยนชื่อแถวที่สร้างครั้งแรกว่า A แล้วให้แต่ละห้องมีหมายเลขเรียงกันไปตามลำดับ โดยให้แต่ละห้องมีหมายเลขอยู่ในวงเล็บดังนี้ A(0), A(1), A(3), ... เรื่อยๆ ไปจนครบ รวมเป็น 11 ห้อง เขาพบว่าเป็นการง่ายเข้าที่จะจำได้ว่าห้อง A(8) มีคนอยู่ 3 คน A(4) มีคนอยู่ 7 คน ฯลฯ



รูปที่ 20.1 เรือนแถว A

กิจการเขารุ่งเรืองขึ้น ต้องสร้างคอนโดมิเนียม 5 ชั้น แต่ละชั้นมีห้อง 8 ห้อง เขาจึงยึดนโยบายเดิมตั้งชื่อห้องแต่ละห้อง โดยมีเลขในวงเล็บตามท้าย 2 ตัว เช่น C(4, 2) ก็หมายถึงเรือนชุด C ชั้นที่ 4 ห้องที่ 2

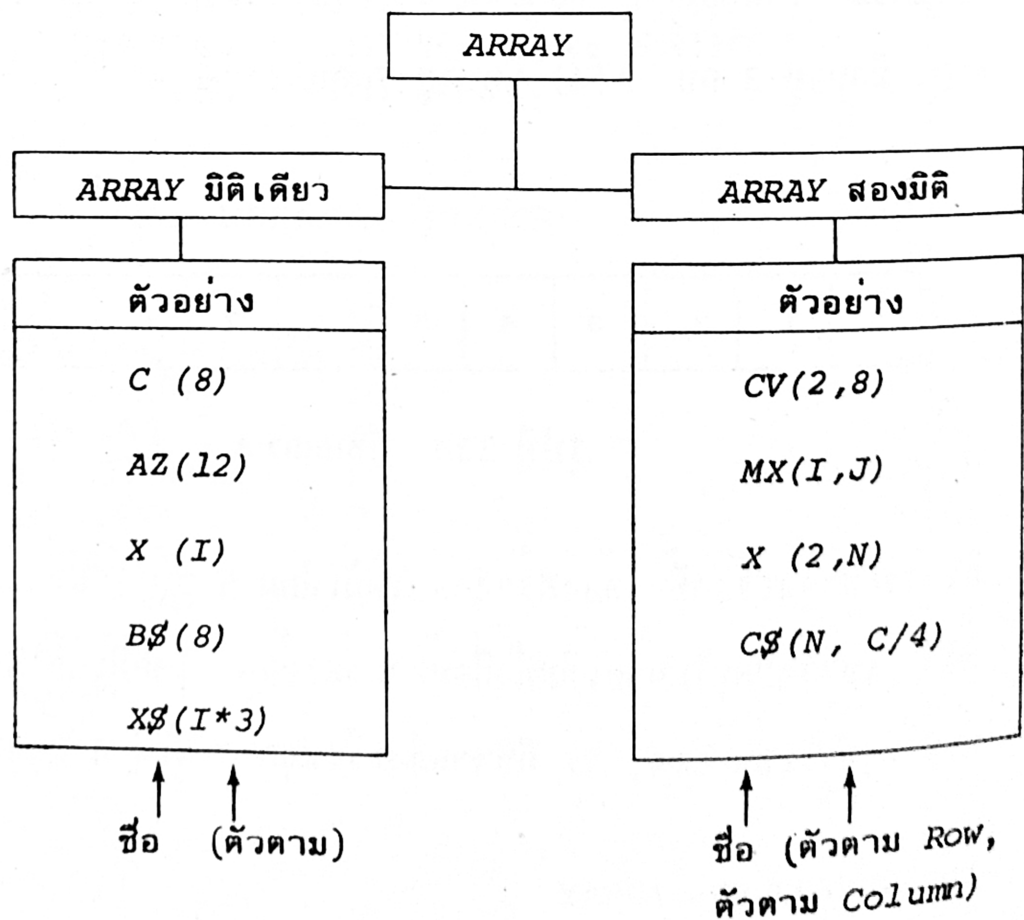
นี่คือความหมายของ ARRAY

ห้องที่

	1	2	3	4	5	6	7	8
ชั้นที่ 5								
ชั้นที่ 4		C(4,2)						
ชั้นที่ 3							C(3,7)	
ชั้นที่ 2								
ชั้นที่ 1								

รูปที่ 20.2 คอนโดมิเนียม 5 ชั้น

20.2 ARRAY คือ ระบบการเรียกตัวแปรที่เป็นหมวดหมู่ เพื่อไม่ให้มีการสับสนกัน ที่ใช้กันมากมีอยู่ 2 แบบ แบบหนึ่งคือ ARRAY แบบมิติเดียว (ONE DIMEN-



SIONAL ARRAY) ประกอบไปด้วยชื่อ (ARRAY NAME) ตามด้วยตัวตาม (SUBSCRIPT) หนึ่งตัวอยู่ในวงเล็บ เปรียบเสมือนหมายเลขห้องของเรือนแถวชั้นเดียว ดังกล่าวมาแล้ว

ARRAY อีกแบบหนึ่งเรียกว่า ARRAY แบบสองมิติ (TWO DIMENSIONAL ARRAY ประกอบไปด้วยชื่อ (ARRAY NAME) ตามด้วยตัวตาม(SUBCRIPT) สองตัวอยู่ในวงเล็บ ตัวตามตัวแรกเป็นหมายเลขแถว(Row) ตัวหลังเป็นหมายเลขสดมภ์(Column) เปรียบเสมือนหมายเลขห้อง และหมายเลขชั้นของอาคารหลายชั้น ดังกล่าวมาแล้ว

ลองดูตัวอย่างง่ายๆ อีกแบบหนึ่ง ARRAY มิติเดียว คือรายการสิ่งของ เช่น รายการอาหารของร้านอาหารแห่งหนึ่งมีดังนี้

รายการที่ 1	แกงไก่
รายการที่ 2	ผัดผัก
รายการที่ 3	แกงจืดหมู

ถ้ารายการอาหารมีเปลี่ยนแปลงตามวัน จะกลายเป็น ARRAY 2 มิติ ดังนี้

รายการ	จันทร์	อังคาร	พุธ
1	แกงไก่	แกงส้ม	แกงป่า
2	ผัดผัก	ยำ	ต้มยำ
3	แกงจืดหมู	ต้มเค็ม	ผัดคะน้า

รูปที่ 20.4 รายการอาหาร

20.3 ชื่อ (ARRAY NAME) มีหลักเกณฑ์การใช้เหมือนกับตัวแปรที่กล่าวในบทที่ 4 ซึ่งอาจจะเป็นตัวแปรที่บรรจุตัวเลข(NUMERICAL VARIABLE) และตัวแปรพันธะ(STRING VARIABLE) ก็ได้

20.4 ตัวตาม(SUBSCRIPT) ของ ARRAY ทั้งสองแบบอาจจะเป็นค่าคงที่ (CONSTANT) เช่น 8, 12 หรือตัวแปร เช่น I, A ก็ได้ หรือยิ่งไปกว่านั้นอาจจะเป็นฟังก์ชั่นทางคณิตศาสตร์ เช่น C/2 ก็ได้

20.5

```
10   DIM A(17)
20   DIM C$(15,38)
```

การใช้ ARRAY จะต้องมีการกำหนดขนาด, ความต้องการไว้ในโปรแกรมด้วยว่าจะต้องมีขนาดเท่าใด(คล้ายๆ กับการเขียนใบจองที่) โดยใช้คำสั่ง DIM ซึ่งย่อมาจาก DIMENSION บรรทัดที่ 10 เป็นการกำหนดขนาด ARRAY มิติเดียว ชื่อ A มีขนาดตั้งแต่ A(0) จนถึง A(17) รวมเป็น 18 หน่วย บรรทัดที่ 20 กำหนดขนาด ARRAY 2 มิติ ว่าจะต้องมีขนาด 16 แถว (0 ถึง 15) และ 39 สดมภ์ (0-38)

20.6 ย้อนกลับไปดูห้องแถวในรูปที่ 20.4 ในเรือนแถว A มีห้องอยู่ 11 ห้อง แต่ละห้องมีคนอยู่มีจำนวนดังนี้

ห้องแรก หรือ A(0) มีคนอยู่ 3 คน
ห้องที่ 2 หรือ A(1) มีคนอยู่ 5 คน
ห้องที่ 3 หรือ A(2) มีคนอยู่ 2 คน

....

ห้องที่ 11 หรือ A(10) มีคนอยู่ 3 คน

เรือนแถว A

ห้องที่	0	1	2	3	4	5	6	7	8	9	10
จำนวนคน	3	5	2	6	7	3	2	2	3	5	3

รูปที่ 20.5 จำนวนคนในเรือนแถว A

```
40  DIM A(10)
50 A(0)  = 3
60 A(1)  = 5

150 A(10) = 3
```

โปรแกรมทางซ้ายมือเป็นตัวอย่างการป้อนข้อมูลเข้า

การเข้าโปรแกรมแบบนี้ เป็นการเยิ่นเย้อ หากเรากำหนดให้ตัวตาม (Subscript) เป็นตัวแปร แล้วใช้ FOR.... NEXT... และ READ.... DATA.... เข้าช่วยจะทำให้ง่ายเข้า ดูโปรแกรมถัดไป

```
5  HOME
10  REM  PRO#20-1
20  REM  ARRAY DEMO
30  DATA 3,5,2,6,7,3,2,2,3,5,3
40  DIM A(10)
50  FOR I = 0 TO 10
60  READ A(I)
70  NEXT I
80  FOR I = 0 TO 10
90  PRINT "A";I,A(I): PRINT
100  NEXT I
110  END
```

```
]RUN
A0          3

A1          5

A2          2

A3          6

A4          7

A5          3

A6          2

A7          2

A8          3

A9          5

A10         3
```

บรรทัดที่ 30 เป็นการนำเอาจำนวนคนที่อาศัยอยู่แต่ละห้องมาเขียนรวมกัน-โดยมีเครื่องหมายคอมม่าคั่น

บรรทัดที่ 40 เป็นการจองที่สำหรับ ARRAY มิติเดียว

บรรทัดที่ 50 ถึง 70 เป็นการบรรจุข้อมูล จำนวนคน ของแต่ละห้องเข้าใน ARRAY แต่ละห้อง โดยการหมุนเป็นวงรอบทั้งหมด 11 รอบ รอบแรก I จะมีค่าเป็นศูนย์ A(I) จะเท่ากับ A(0) และจะอ่านค่าจาก DATA ตัวแรก คือ 3 ซึ่งจะรับเข้ามาบรรจุไว้

ในรอบที่ 2 I จะมีค่าเป็น 1 A(1) จะรับ DATA ตัวที่ 2 คือ 5 เข้าไว้ตามลำดับ เมื่อครบ 11 รอบ จะมาเข้าวงรอบต่อมา ซึ่งจะพิมพ์ข้อมูล คือ อักษร A ตัวแปร ตั้งแต่ศูนย์จนถึง 10 และค่าของตัวคงที่ที่ถูกบรรจุไว้ใน ARRAY ในวงจรรอบ

โปรแกรมนี้ อาจจะทำให้ง่ายเข้าด้วยการตัดวงรอบหลังออกโดยเอาบรรทัดที่ 90 ไปบรรจุไว้เป็นส่วนหนึ่งของวงรอบแรก ดังในโปรแกรมที่ 20-2

```
5  HOME
10  REM   PRO#20-2
20  REM   ARRAY DEMO 2
30  DATA 3,5,2,6,7,3,2,2,3,5,3
40  DIM A(10)
45 N = 0
46  PRINT "ROOM NO.","OCCUPANTS":
      PRINT
50  FOR I = 0 TO 10
60  READ A(I)
70  PRINT "A";I,A(I)
80 N = N + A(I)
90  NEXT I
100  PRINT "TOTAL OCCUPANTS= ";N
110  PRINT "RENTAL PER OCCUPANT=5
     0 BAHT/MO"
120  PRINT "TOTAL RENTAL = ";N *
     50;" BAHT"
130  END

]RUN
ROOM NO.          OCCUPANTS

A0                3
A1                5
A2                2
A3                6
A4                7
A5                3
A6                2
A7                2
A8                3
A9                5
A10               3
TOTAL OCCUPANTS= 41
RENTAL PER OCCUPANT=50 BAHT/MO
TOTAL RENTAL = 2050 BAHT
```

บรรทัดที่ 80 จะเป็นการรวมจำนวนผู้มาพักทั้งหมด ซึ่งจะแสดงจำนวนออกมาในบรรทัดที่ 100 เมื่อนำมาคูณกับราคาต่อหัวก็จะได้ราคาค่าเช่าทั้งสิ้น

	1	2	3	4	5	6	7	8
ชั้นที่ 5	3	2	1		5	2	3	6
ชั้นที่ 4	4	5	2	1	3	1	5	2
ชั้นที่ 3	8	4	9	3	3	1	5	2
ชั้นที่ 2	6		5	3		2	5	1
ชั้นที่ 1	3	4	2	1	5	5	4	5

รูปที่ 20.5

รูปที่ 20.5 แสดงหอพัก 5 ชั้น แต่ละชั้นมี 8 ห้อง ตัวเลขที่อยู่ในแต่ละห้อง คือ จำนวนคนที่พักอยู่

ศึกษาโปรแกรมที่ 20-3 ซึ่งเป็นการใช้ ARRAY สองมิติ

```
5  HOME
10  REM  PRO#20-3
20  REM  TWO DIMENSION ARRAY
30  DIM ROOM(5,8)
40  FOR FL = 1 TO 5
50  FOR R = 1 TO 8
60  READ ROOM(FL,R)
70 N(FL) = N(FL) + ROOM(FL,R)
80  NEXT R
90  PRINT "NO.OF OCCUPANTS IN ";F
    L;" FLOOR= ";N(FL)
100 SUM = SUM + N(FL)
110  NEXT FL
120  PRINT "TOTAL OCCUPANTS= ";SU
     M: PRINT
130  PRINT " WANT TO FIND NO. OF
     OCCUPANTS IN ANY ROOM ? PRIN
     T Y OR N"
140  INPUT Y$
150  IF Y$ < > "Y" THEN END
160  INPUT "WHICH FLOOR? ";FL
170  INPUT "WHICH ROOM ? ";R
```

```
180  PRINT " NO. OF OCCUPANTS = "
     ;ROOM(FL,R)
190  PRINT : PRINT : GOTO 130
200  DATA  3,4,2,1,5,5,4,5
300  DATA  6,0,5,3,0,2,5,1
400  DATA  8,4,9,3,3,1,5,2
500  DATA  4,5,2,1,3,1,5,2
600  DATA   3,2,1,0,5,2,3,6

]RUN
NO.OF OCCUPANTS IN 1 FLOOR= 29
NO.OF OCCUPANTS IN 2 FLOOR= 22
NO.OF OCCUPANTS IN 3 FLOOR= 35
NO.OF OCCUPANTS IN 4 FLOOR= 23
NO.OF OCCUPANTS IN 5 FLOOR= 22
TOTAL OCCUPANTS= 131

 WANT TO FIND NO. OF OCCUPANTS IN ANY ROOM ?
   PRINT Y OR N
?Y
WHICH FLOOR? 3
WHICH ROOM ? 1
 NO. OF OCCUPANTS = 8

 WANT TO FIND NO. OF OCCUPANTS IN ANY ROOM ?
   PRINT Y OR N
?Y
WHICH FLOOR? 5
WHICH ROOM ? 8
 NO. OF OCCUPANTS = 6
```

บรรทัดที่ 30 เป็นการกำหนดขอบเขตของ ARRAY ชื่อของ ARRAY เราใช้ ROOM ตัวตามใช้ FL ซึ่งเป็นตัวแปรแทนชั้น และ R แทนห้อง

บรรทัดที่ 40 ถึง 110 เป็นวงรอบซ้อน(NESTED LOOP) คือ วงรอบนอกเป็นเรื่องของชั้น วงรอบในบรรทัดที่ 50 ถึง 80 เป็นเรื่องของห้องในแต่ละชั้น

เมื่อครั้งทำงานผ่านบรรทัดที่ 40 ครั้งแรก FL จะเท่ากับ 1 แล้วผ่านไปวงรอบในครั้งแรก R จะเท่ากับ 1 บรรทัดที่ 60 เป็นการบรรจุ DATA ตัวแรกคือ เลข 3 ในบรรทัดที่ 300 บรรทัดที่ 70 จะเป็น ARRAY มิติเดียวซึ่งจะทำหน้าที่รวบรวมจำนวนคนที่อยู่ในห้องแต่ละชั้น ซึ่งในครั้งแรก FL=1 N(FL) จะเป็น N(1) ซึ่งเมื่อวงรอบใน ครบ 8 ครั้งแล้ว จะผ่านมาแสดงผล คือจำนวนคนในแต่ละชั้น บรรทัดที่ 100 จะรวมคนในทุกชั้นไว้อีก เมื่อโปรแกรมถึงบรรทัดที่ 110 จะกลับไปที่บรรทัดที่ 40 FL จะเปลี่ยนเป็น 2 คือชั้นที่ 2 แล้วเข้าวงรอบในอีก จนกระทั่งวงรอบในครบ 8 รอบ และวงรอบนอกครบ 5 รอบ ข้อมูลทั้งหมดก็จะถูกนำไปบรรจุใน ARRAY ตามหมายเลข ซึ่งจากนี้เราสามารถเรียกข้อมูลที่บรรจุใน ARRAY ใดมาใช้ก็ได้

20.8

	ห้องที่ 1	ห้องที่ 2	ห้องที่ 3
ชั้นที่ 1	E H	NUEY	NENG
ชั้นที่ 1	CHAN	WAN	ORH

รูปที่ 20.6

สิ่งที่บรรจุใน ARRAY แทนที่จะเป็นตัวเลข เราอาจให้เป็นตัวอักษรก็ได้ ในกรณีเช่นนี้ ชื่อของ ARRAY จะต้องเป็นตัวแปรพันธะ(STRING VARIABLE)

```
5  HOME
10  REM  PRO#20-4
20  DIM C$(2,3)
30  FOR F = 1 TO 2
40  FOR B = 1 TO 3
50  READ C$(F,B)
60  NEXT B,F
70  INPUT "WHICH FLOOR ?";F
```

```
80  INPUT "WHICH ROOM ?";B
90  PRINT "NAME OF OCCUPANT-";C$(
     F,B): PRINT
100  GOTO 70
200  DATA  CHAN,WAN,ORH,EH,NUEY,N
     ENG

]RUN
WHICH FLOOR ?1
WHICH ROOM ?3
NAME OF OCCUPANT-ORH

WHICH FLOOR ?2
WHICH ROOM ?2
NAME OF OCCUPANT-NUEY
```

ในโปรแกรม 20-4 โปรดสังเกตบรรทัดที่ 60 เราอาจจะนำ NEXT ของวงรอบซ้อนมาเขียนรวมกันได้ โดยใช้เครื่องหมายคอมม่าคั่น

บรรทัดที่ 20 ถึง 60 เป็นตัวอย่างที่เป็นมาตรฐาน เมื่อจะใช้ ARRAY 2 มิติ

20.9 โปรแกรม 20-5 เป็นโปรแกรมสำหรับจัดลำดับตัวเลขจากน้อยไปหามากซึ่งเป็นโปรแกรมที่น่าสนใจที่น่าศึกษามาก โปรแกรมตั้งแต่บรรทัดแรก จนถึ

```
5  HOME
10  REM  PRO#20-5
20  REM  NUMBER'S ORDER
30  DIM A(6)
50  PRINT "GIVE ME ANY SEVEN NUMB
     ERS"
60  PRINT "I WILL ARRANGE THE ORD
     ER"
65  FOR I = 0 TO 6
70  INPUT A(I)
80  NEXT I
90 N = 0
100  FOR I = 0 TO 5
```

```
110  IF A(I) <  = A(I + 1) THEN  GOTO
     160
120 M = A(I)
130 A(I) = A(I + 1)
140 A(I + 1) = M
150 N = 1
160  NEXT I
170  IF N = 1 THEN 90
175  PRINT "HERE IS THE NUMBER IN
      ORDER"
180  FOR I = 0 TO 6
190  PRINT A(I)
200  NEXT I


]RUN
GIVE ME ANY SEVEN NUMBERS
I WILL ARRANGE THE ORDER
?45
?3452
?9
?67500
?786
?567
?98

HERE IS THE NUMBER IN ORDER

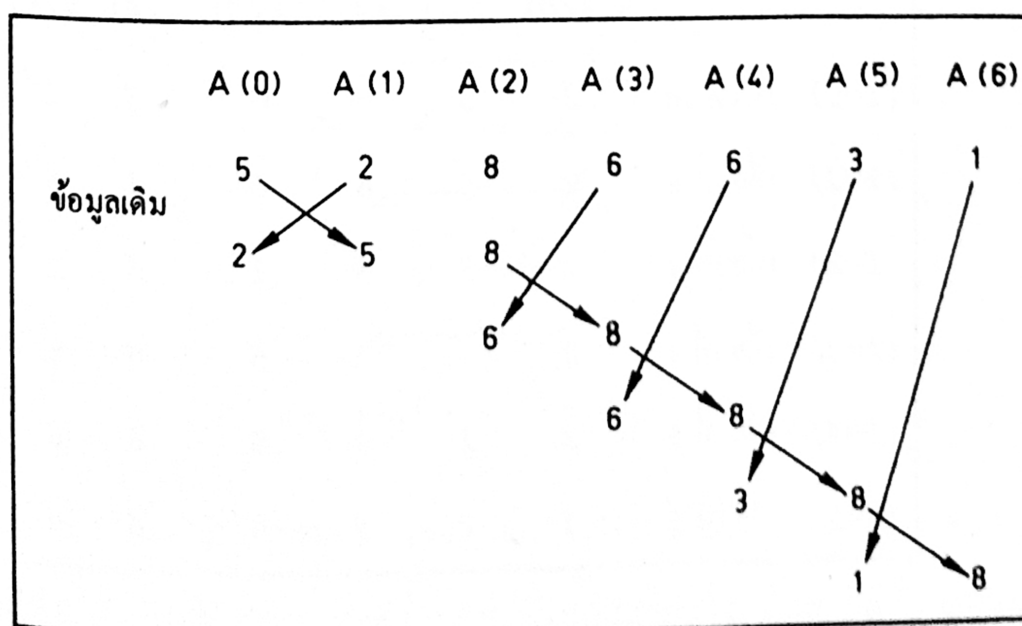
9
45
98
567
786
3452
67500
```

บรรทัดที่ 80 เป็นการป้อนตัวเลขคละไม่เรียงลำดับ เข้าไปอยู่ใน ARRAY บรรทัดที่ 90 ถึง 170 เป็นการจัดลำดับเลขให้เรียงจากน้อยไปหามาก ซึ่งเป็นขบวนการที่ซับซ้อนพอสมควร ตัว N ในบรรทัดที่ 90 และ 150 เป็นตัวบ่งชี้ว่าขบวนการจะสิ้นสุดเมื่อใด บรรทัดที่ 120 ถึง 140 เป็นการสลับตัว

เลขระหว่าง 2 หน่วย ของ ARRAY ซึ่งจะทำได้โดยการกำหนดให้มีตัวแปรนอก เข้ามาช่วยเป็นที่พัก (สมมติว่ามีถัง 2 ใบ ใบที่ 1 บรรจุน้ำ ใบที่ 2 บรรจุน้ำมัน ถ้าจะเปลี่ยนถังกัน ก็จำเป็นจะต้องมีถังเปล่าใบที่ 3 เอาน้ำในถังที่ 1 ใส่ไปในถังที่ 3 แล้วเอาน้ำมันในถังที่ 2 ไปใส่ในถังที่ 1 ที่ว่างแล้วจึงเอาน้ำในถังที่ 3 มาใส่ในถังที่ 2 เป็นอันเสร็จ)

เพื่อให้ง่ายต่อการอธิบาย เรามาสมมติว่าตัวเลขที่ป้อนไปยังหน่วยของ ARRAY มีดังนี้ 5, 2, 8, 6, 6, 3, 1

เมื่อเครื่องดำเนินมาถึงบรรทัดที่ 90 N จะมีค่าเป็นศูนย์ เมื่อเข้ามาถึงบรรทัดที่ 100 จะเข้าวงรอบ FOR-NEXT ในรอบแรกเป็นการเปรียบเทียบค่าของ A(0) กับ A(0+1) หรือ A(1) ถ้า A(0) น้อยกว่า หรือเท่ากับ A(1) จะกระโดดมาที่บรรทัดที่ 160 แต่ถ้า A(0) มีค่ามากกว่า A(1) จะมีการแลกค่ากันระหว่าง A(0) และ A(1)โดยจะเอาค่าที่มากมาไว้ที่ A(1) ตามตัวอย่างของเรา A(0) มากกว่า A(1) จึงมีการแลกกัน โดยเอา 5 มาไว้ใน A(1) และเอา 2 มาไว้ใน A(0) โปรดสังเกตว่า N จะเปลี่ยน



ค่าเป็น 1 ในบรรทัดที่ 150 ในวงรอบ FOR-NEXT รอบที่ 2 ค่า I จะเท่ากับ 1 จึงมีการทดสอบระหว่าง A(1) กับ A(1+1) หรือ A(2) ตามตัวอย่างของเรา A(1) มีค่าน้อยกว่า A(2) จึงไม่มีการแลกเปลี่ยนกับค่าของ N ยังคงเป็น 1 อยู่

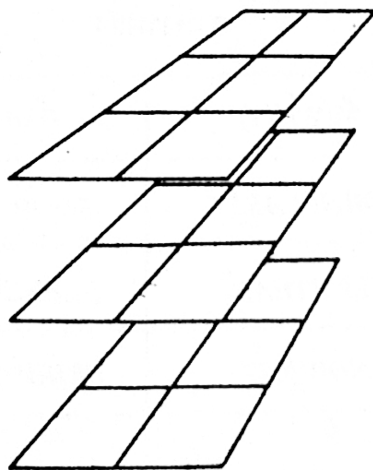
ในวงรอบที่ 3 เปรียบเทียบค่าใน A(2) กับค่าใน A(3) ค่าใน A(2) คือ 8 มากกว่าค่าใน A(3) ซึ่งเท่ากับ 6 จึงต้องมีการเปลี่ยนเลขกันอีก ทำดังนี้เรื่อยไปจนครบ 6 เที่ยว พึงสังเกตว่าค่า 8 ตามตัวอย่างซึ่งเป็นค่าที่สูงที่สุด จะเลื่อนมาจนกระทั่งมาอยู่ใน A(6) เป็นอันจบวงรอบ FOR-NEXT เที่ยวแรก ขณะนี้ N ยังมีค่าเป็น 1 อยู่ เมื่อมาถึงบรรทัดที่ 160 ก็ถูกสั่งให้เริ่มวงจร FOR-NEXT เที่ยวที่ 2 แต่ก่อนอื่นต้องเปลี่ยนค่า N เป็น 0 เสียก่อน เมื่อจบวงรอบ FOR-NEXT เที่ยวที่ 2 ค่าของตัวเลขที่อยู่ใน ARRAY ตัวที่มีค่ามากที่สุด จะถูกเลื่อนไปจนอยู่ใน A(5) เป็นดังนี้เรื่อยไป

		A(0)	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)
(I=0)	เที่ยวแรก	2	5	6	6	3	1	8
(I=1)	เที่ยวที่ 2	2	5	6	3	1	6	8
(I=2)	เที่ยวที่ 3	2	5	3	1	6	6	8
(I=3)	เที่ยวที่ 4	2	3	1	5	6	6	8
(I=4)	เที่ยวที่ 5	2	1	3	5	6	6	8
(I=5)	เที่ยวที่ 6	1	2	3	5	6	6	8

ในวงรอบ FOR-NEXT เที่ยวที่ 6 ตัวเลขจะเรียงลำดับกัน จากน้อยไปหามาก แต่เมื่อ N ยังเป็น 1 อยู่ โปรแกรมจะถูกสั่งให้กลับไปบรรทัดที่ 90 เพื่อทำ N ให้เป็น 0 ในเที่ยวนี้ ไม่มีการแลกเปลี่ยนตัวเลขกัน โปรแกรมจึงไม่ผ่านบรรทัดที่ 150 ค่าของ N ยังคงเป็น 0 อยู่ จึงผ่านบรรทัดที่ 170 ไปบรรทัดที่ 175 ได้ และแสดงค่าตัวเลขใน ARRAY ซึ่งได้เรียงลำดับไว้แล้ว

หากต้องการให้แสดงตัวเลขจากมากไปหาน้อย จะทำได้ 2 วิธีคือ วิธีหนึ่งเปลี่ยนเครื่องหมายในบรรทัดที่ 110 เป็น > (มากกว่า)หรือเปลี่ยนบรรทัดที่ 180 เป็น FOR I = 6 TO 0 STEP - 1

20.10 ตึกหลังหนึ่งมี 3 ชั้น แต่ละชั้นมี 2 แถว แถวละ 3 ห้อง นี่คือ ARRAY 3 มิติ ลองย้อนกลับไปดูรายการอาหารใน 20.2 หากมีรายการอาหารฝรั่ง อาหารแขก ฯลฯ มาอีกโดยที่แต่ละรายการมี อาหาร 3 รายการและแต่ละวันไม่ซ้ำกัน นี่ก็คือ ARRAY 3 มิติ



สรุป ARRAY คือ ระบบการรวมและการเรียกตัวแปรที่เป็นหมวดหมู่ ARRAY อาจจะเรียกว่า SUBSCRIPT VARIABLE ที่ใช้กันมากมี 2 แบบ คือ ARRAY

แบบมิติเดียว และ ARRAY สองมิติ แต่อาจจะเป็นหลายๆ มิติ ได้โดยการเพิ่มจำนวนตัวเลขในวงเล็บข้างหลัง การบรรจุตัวเลข หรือตัวอักษรลงใน ARRAY วิธีที่สะดวกที่สุดก็โดยการใช้คำสั่ง FOR... NEXT

แบบฝึกหัดที่ 20

20.1 จงเขียน ARRAY 3 มิติ โดยใช้รายการอาหารรวม 3 ชาติ 3 ชนิด และ 3 วัน ดังนี้

อาหารไทย			
รายการ	จันทร์ (1)	อังคาร (2)	พุธ (3)
(1) เช้า	แกงไก่	ข้าวผัด	กุ้งทอด
(2) กลางวัน	ผัดถั่วงอก	ยำสลัด	แกงเนื้อ
(3) เย็น	แกงจืดหมู	ไข่ดาว	ผัดเปรี้ยวหวาน

อาหารฝรั่ง			
รายการ	จันทร์ (1)	อังคาร (2)	พุธ (3)
1. เช้า	*CORNFLAKES*	*SOUP*	*COTTAGE PIE*
2. กลางวัน	*BEEF STEAK*	*PEPPER STEAK*	*PORK CHOP*
3. เย็น	*SANDWICH*	*FRIED CHICKEN*	*COLD BEEF*

เฉลยแบบฝึกหัดบทที่ 20

20.1

```
5  REM  S20.1
10  HOME
20  DIM F$(2,3,3)
30  FOR D = 1 TO 2
40  FOR I = 1 TO 3
50  FOR K = 1 TO 3
60  READ F$(D,I,K)
70  NEXT K,I,D
80  INPUT "YOUR CHOICE*** 1=THAI
     FOOD, 2=EUROPEAN FOOD***  1
     OR 2 ?  ";D: PRINT
85  IF D > 2 GOTO 80
90  INPUT "WHICH MEAL - 1=BREAKFA
     ST, 2=LUNCH,  3=DINNER - 1 O
     R 2 OR 3 ?  ";I: PRINT
95  IF I > 3 GOTO 80
100  INPUT "WHAT DAY ## 1=MONDAY,
      2=TUESDAY, 3=WEDNESDAY ## 1
      OR 2 OR 3 ?  ";K: PRINT
105  IF K > 3 GOTO 100
110  PRINT
120  FLASH : PRINT F$(D,I,K): NORMAL

130  DATA   CHICKEN CURRY,FRIED R
     ICE,FRIED SHRIMPS,CHOP SUEY,
     SALAD,BEEF CURRY,PORK SOUP,F
     RIED EGGS,SWEET & SOUR
140  DATA  CORNFLAKES,SOUP,COTTAG
     E PIE,BEEF STEAK,PEPPER STEA
     K,PORK CHOP,SANDWITCH,FRIED
     FISH,COLD BEEF
150  PRINT : PRINT : INPUT "WANT
     TO LOOK AT THE MENU ? Y OR N
       ";A$
160  IF A$ <  > "Y" THEN  END
170  HOME : PRINT  TAB( 10)"HERE
     IS THE MENU"
180  PRINT : SPEED= 100
190  PRINT  TAB( 15)"MONDAY"
200  GOSUB 3000
```

```
220 B$ = "BREAKFAST":C$ = "LUNCH"
    :D$ = "DINNER"
230  PRINT B$,F$(1,1,1)
240  PRINT C$,F$(1,2,1)
250  PRINT D$,F$(1,3,1)
260  GOSUB 4000
270  PRINT B$,F$(2,1,1)
280  PRINT C$,F$(2,2,1)
290  PRINT D$,F$(2,3,1)
300  PRINT : PRINT  TAB( 15)"TUES
     DAY": PRINT
310  GOSUB 3000
320  PRINT B$,F$(1,1,2)
330  PRINT C$,F$(1,2,2)
340  PRINT D$,F$(1,3,2)
350  GOSUB 4000
360  PRINT B$,F$(2,1,2)
370  PRINT C$,F$(2,2,2)
380  PRINT D$,F$(2,3,2)
390  PRINT : PRINT  TAB( 15)"WEDN
     ESDAY": PRINT
400  GOSUB 3000
410  PRINT B$,F$(1,1,3)
420  PRINT C$,F$(1,2,3)
430  PRINT D$,F$(1,3,3)
440  GOSUB 4000
450  PRINT B$,F$(2,1,3)
460  PRINT C$,F$(2,2,3)
470  PRINT D$,F$(2,3,3)
475  SPEED= 255
480  PRINT : PRINT : PRINT  TAB(
     15)"ENJOY YOUR MEAL": NORMAL
     : END
3000  PRINT : PRINT " ","THAI"
3010  PRINT : PRINT " ","----"
3020  RETURN
4000  PRINT : PRINT " ","FARANG"
4010  PRINT : PRINT " ","------"
4020  RETURN

]RUN
YOUR CHOICE*** 1=THAI FOOD, 2=EUROPEAN FOOD***
     1 OR 2 ?  1
```

```
WHICH MEAL - 1=BREAKFAST, 2=LUNCH, 3=DINNER - 1
      OR 2 OR 3 ?  3
WHAT DAY ## 1=MONDAY, 2=TUESDAY, 3=WEDNESDAY
      ## 1 OR 2 OR 3 ?  2


FRIED EGGS


WANT TO LOOK AT THE MENU ? Y OR N  Y
            HERE IS THE MENU

                  MONDAY

                    THAI

                    ----
BREAKFAST           CHICKEN CURRY
LUNCH               CHOP SUEY
DINNER              PORK SOUP

                    FARANG

                    -----
BREAKFAST           CORNFLAKES
LUNCH               BEEF STEAK
DINNER              SANDWITCH

                  TUESDAY


                    THAI

                    ----
BREAKFAST           FRIED RICE
LUNCH               SALAD
DINNER              FRIED EGGS

                    FARANG

                    -----
```

BREAKFAST	SOUP
LUNCH	PEPPER STEAK
DINNER	FRIED FISH

WEDNESDAY

THAI

BREAKFAST	FRIED SHRIMPS
LUNCH	BEEF CURRY
DINNER	SWEET & SOUR

FARANG

BREAKFAST	COTTAGE PIE
LUNCH	PORK CHOP
DINNER	COLD BEEF

ENJOY YOUR MEAL

21

INT, ABS, RND

21.1 INT ย่อมาจากคำว่า INTEGER หรือจำนวนเต็มที่ได้มาจากการตัดเศษ ถ้าเลขจำนวนนั้นเป็นบวก

ตัวอย่าง X = INT (4.05)

X จะเท่ากับ 4

```
5  REM  PRO#21-1
10  HOME
20  PRINT  TAB( 5)"INT(X) DEMO.":
      PRINT
30  PRINT "A","INT(A)"
35  PRINT "---","---": PRINT
40  FOR I = 1 TO 6
50  READ A
60  PRINT A, INT (A)
70  NEXT I
80  DATA  4.05,4.85,-13.2,-13.8,.
     5,.05

]RUN
     INT(X) DEMO.

A                    INT(A)
---                  ---

4.05                 4
4.85                 4
-13.2                -14
-13.8                -14
.5                   0
.05                  0
```

ในกรณีที่เลขจำนวนนั้นเป็นลบ ค่าที่ได้จากการ INT(X) จะเป็นดังตัวอย่างนี้

X = INT (-13.2)

X จะเท่ากับ -14

หลัง INT จะต้องมีตัวคงที่ หรือฟังก์ชั่นทางคณิตศาสตร์ ซึ่งอยู่ในวงเล็บตาม

โปรแกรมนี้แสดงผลของการ INT(X) ตัวเลขต่างๆ

21.2 โปรแกรม 21-2 จะนำ INT(X) มาใช้ในการปัดทศนิยม ตามตำแหน่งที่ต้องการ

```
5  HOME
10  REM   PRO#21-2
20  PRINT  TAB( 5)"USE OF INT( )
                 ": PRINT
30  PRINT "A-REAL","A-2 DECIMALS"
     : PRINT
40  FOR I = 1 TO 4
50  READ A
60 B =  INT (100 * A + .5) / 100
70  PRINT A,B
80  NEXT I
90  DATA    13.2215,5.1251,13.244
     1,4.5361

]RUN
    USE OF INT( )

A-REAL          A-2 DECIMALS

13.2215         13.22
5.1251          5.13
13.2441         13.24
4.5361          4.54
```

โปรดสังเกตการปัดเศษของเลขแต่ละจำนวน

โปรแกรมนี้สำคัญอยู่ที่บรรทัดที่ 60 ซึ่งทำงานดังนี้

สมมติว่า A	=	5.1251
100*A	=	512.51
100*A+.5	=	513.01
INT(100*A+.5)	=	513
INT(100*A+.5)/100	=	5.13
ถ้า A	=	13.2441
100*A	=	1324.41
100*A+.5	=	1324.91
INT(100*A+.5)	=	1324
INT(100*A+.5)/100	=	13.24

หลักการมีดังนี้ เราเอา 100 คูณ เพื่อเลื่อนทศนิยมตัวที่ 2 เป็นหลักหน่วย แล้วเอา 0.5 มาบวก หากทศนิยมตัวแรกเท่ากับหรือมากกว่า 5 หลักหน่วยจะเพิ่มขึ้น 1 หน่วย ถ้าทศนิยมตัวแรกน้อยกว่า 5 หลักหน่วยจะไม่เพิ่ม เมื่อทำการตัดเศษทิ้งแล้วหารด้วย 100 ก็จะได้ การปัดเศษตามต้องการ

ถ้าต้องการปัดเศษทศนิยมตัวที่ 3 ก็เปลี่ยนบรรทัดที่ 60 เป็น

```
60 B  =  INT(1000*A+.5)/1000
```

21.3 ABS(S) คือ ABSOLUTE VALUE OF S คือ ค่าที่เป็นบวกของเลข

21.4 RND(1) เป็นคำย่อของ RANDOM NUMBER OF X คือ ตัวเลขที่เครื่องสุ่มขึ้นมา โดยไม่มีวิธีการที่แน่นอน

```
5  REM  PRO#21-3
20  PRINT "RANDOM NUMBERS": PRINT

30  FOR I = 1 TO 10
40  PRINT  RND (1),
50  NEXT
60  END

]RUN
RANDOM NUMBERS

.175432536         .634381771
.374943727         .8122274
.0907438741        .51012233
.784910901         .469839189
.895832578         .689005174

]RUN
RANDOM NUMBERS

.889798759         .649576523
.89646178          .318356969
.550121421         .892336952
.225592532         .7912905
.216543946         .43363092

]RUN
RANDOM NUMBERS

.282056189         .369181561
.983754785         .622354549
.494630091         .243440312
.943403345         .812532644
.247210001         .692361516
```

โปรแกรมที่ 21-3 เมื่อ RUN ออกมาแล้ว จะได้ตัวเลขที่เป็นทศนิยม 9 ตำแหน่ง มา 10 ตัว โดยไม่ซ้ำกันเลย ลองกด RUN ดูสัก 2-3 ครั้ง แล้วเปรียบเทียบตัวเลขกันดู (โปรดสังเกตว่าโปรแกรมนี้ไม่มี HOME เพราะต้องการเปรียบเทียบตัวเลขกัน)

21.5 ในกรณีที่จะนำตัวเลขที่ได้จากการสุ่มไปใช้ มักจะเป็นเลขจำนวนเต็มมากกว่า สมมติว่าต้องการตัวเลขสุ่มที่มีค่าระหว่าง 1 ถึง 6 จะเขียนโปรแกรมได้ตามโปรแกรม 21-4

```
5  REM  PRO#21-4
20  PRINT "SIMPLE RANDOM FIGURES"
    : PRINT
30  FOR I = 1 TO 6
40  PRINT  INT (6 *  RND (1)) + 1
    ,
50  NEXT I
60  END

]RUN
SIMPLE RANDOM FIGURES

6                4                5
6                5                3

]RUN
SIMPLE RANDOM FIGURES

3                2                1
3                4                5

]RUN
SIMPLE RANDOM FIGURES

6                4                4
2                4                1
```

บรรทัดที่ 40 อันเป็นหัวใจของโปรแกรมนี้ ทำงานดังนี้ ตัวเลขที่ได้จาก RND(1) แต่ละตัวจะเป็นเลขทศนิยม 9 ตำแหน่งและมีค่าน้อยกว่า 1 เมื่อเอา 6 คูณ ค่าที่ได้จะอยู่ระหว่างทศนิยมที่ต่ำกว่าหนึ่งและ 5 กว่าๆ เมื่อ INT ออกมา ค่าจะได้ตั้งแต่ 0 ถึง 5 เมื่อเอา 1 บวกเข้าไปในทุกค่าก็จะได้ตัวเลขที่มีค่าที่อยู่ระหว่างจำนวนเต็ม 1 ถึง 6

21.6 โปรแกรมนี้เครื่องจะสุ่มตัวเลขตั้งแต่ 1 ถึง 10 มาแล้วให้เราทาย บรรทัดที่ 30 จะกำหนดจำนวนสูงสุด ซึ่งเป็นจำนวนเต็มที่ ที่เครื่องจะสุ่มมา หากต้องการให้เครื่องสุ่มตัวเลขตั้งแต่ 1 ถึง 100 ก็เพียงแต่เปลี่ยนตัวเลข 10 ในบรรทัดที่ 30 เป็น 100 เท่านั้น

```
5  REM  PRO#21-5
10  REM *PICK A NUMBER*
20  HOME :M = 0
30 X =  INT (10 *  RND (1)) + 1
35  PRINT "I HAVE A SPECIFIC NUMB
     ER IN MIND": PRINT
40  INPUT "ENTER ANY NUMBER BETWE
     EN 1 & 10   ";N
50  PRINT : PRINT :M = M + 1
60  IF N = X THEN 90
70  IF N > X THEN 110
80  GOTO 120
90  FOR I = 1 TO 5: PRINT  CHR$ (
     7)
95  NEXT : FLASH : PRINT "YOU HIT
      THE TARGET BY ";M;" TRIAL"
96  NORMAL
100  FOR I = 1 TO 5000: NEXT : GOTO
     20
110  INVERSE : PRINT " TOO HIGH":
      NORMAL : PRINT : PRINT : GOTO
     40
120  INVERSE : PRINT " TOO LOW": NORMAL
      : PRINT : PRINT : GOTO 40

]RUN
I HAVE A SPECIFIC NUMBER IN MIND

ENTER ANY NUMBER BETWEEN 1 & 10   5


 TOO HIGH


ENTER ANY NUMBER BETWEEN 1 & 10   1
```

```
 TOO LOW

ENTER ANY NUMBER BETWEEN 1 & 10    4

 TOO HIGH

ENTER ANY NUMBER BETWEEN 1 & 10    3

 TOO HIGH

ENTER ANY NUMBER BETWEEN 1 & 10    2
```

21.7 โปรแกรมม้าแข่ง

```
5  REM  PRO#21-6
10  HOME
30  VTAB 10: SPEED= 100: GOSUB 50
     00: PRINT : PRINT : SPEED= 3
     0
40  PRINT "          H O R S E   R A
      C I N G": PRINT : PRINT
50  SPEED= 180: PRINT "       PROGR
     AMMED BY PRAPAT & SON": PRINT
     : PRINT
60  SPEED= 200: GOSUB 6000: SPEED=
     200
70  HOME :S$ = " "
80 A$ = ">>>--1-->":B$ = ">>>--2-
     ->":C$ = ">>>--3-->"
90 I = 0:M = 0:N = 0
100  VTAB 5: PRINT A$: VTAB 10: PRINT
     B$: VTAB 15: PRINT C$
110  VTAB 20: PRINT "PRESS 'S' TO
      START": GET D$
150 D =  INT (3 *  RND (1)) + 1
160  ON D GOTO 1000,2000,3000
```

```
1000  VTAB 5:I = I + 1
1010  HTAB I: PRINT S$ + A$
1040  IF I = 28 THEN  FLASH : PRINT
     "WIN": NORMAL : GOTO 4000
1050  GOTO 150
2000  VTAB 10:M = M + 1
2010  HTAB M: PRINT S$ + B$
2020  IF M = 28 THEN  FLASH : PRINT
     "WIN": NORMAL : GOTO 4000
2030  GOTO 150
3000  VTAB 15:N = N + 1
3020  HTAB N: PRINT S$ + C$
3030  IF N = 28 THEN  FLASH : PRINT
     "WIN": NORMAL : GOTO 4000
3040  GOTO 150
4000  PRINT  CHR$ (7): VTAB (20):
      PRINT "AGAIN ? PRESS 'Y OR
     N'": GET E$
4020  IF E$ <  > "N" THEN 70
4030  SPEED= 255: HOME : END
5000  FOR I = 1 TO 40: PRINT "*";

5010  NEXT : RETURN
6000  FOR I = 1 TO 40: PRINT "*";
      CHR$ (7);
6010  NEXT : RETURN

]RUN
****************************************

       H O R S E   R A C I N G


     PROGRAMMED BY PRAPAT & SON


****************************************
```

```
PRESS 'S' TO START
 >>>--3-->
 >>>--2-->
  >>>--2-->
 >>>--1-->
   >>>--2-->
  >>>--3-->
    >>>--2-->
  >>>--1-->
     >>>--2-->
      >>>--2-->
       >>>--2-->
   >>>--3-->
    >>>--3-->
     >>>--3-->
   >>>--1-->
        >>>--2-->
       >>>--3-->
          >>>--2-->
     >>>--1-->
           >>>--2-->
      >>>--1-->
       >>>--1-->
        >>>--3-->
        >>>--1-->
         >>>--3-->
         >>>--1-->
          >>>--3-->
          >>>--1-->
           >>>--1-->
           >>>--3-->
            >>>--1-->
            >>>--2-->
             >>>--2-->
              >>>--2-->
            >>>--3-->
             >>>--3-->
               >>>--2-->
                >>>--2-->
             >>>--1-->
              >>>--1-->
              >>>--3-->
                 >>>--2-->
               >>>--1-->
```

```
             >>>--3-->
                >>>--2-->
              >>>--3-->
               >>>--3-->
                >>>--3-->
                 >>>--3-->
                 >>>--2-->
                  >>>--2-->
              >>>--1-->
                  >>>--3-->
               >>>--1-->
                   >>>--3-->
                >>>--1-->
                 >>>--1-->
                  >>>--1-->
                    >>>--3-->
                   >>>--2-->
                   >>>--1-->
                    >>>--1-->
                     >>>--3-->
                      >>>--3-->
                    >>>--2-->
                     >>>--1-->
                       >>>--3-->
                     >>>--2-->
                      >>>--2-->
                        >>>--3-->
                      >>>--1-->
                         >>>--3-->
                       >>>--1-->
                        >>>--1-->
                       >>>--2-->
                          >>>--3-->
                           >>>--3-->
WIN
```

21.8 อาจจะมีผู้สงสัยว่าตัวเลขที่อยู่ในวงเล็บที่ตามหลัง RND นั้น จะใช้ตัวเลขอื่นนอกจาก 1 จะได้ไหม ตอบว่าได้ และผลที่ได้จากการ RND จะเป็นรูปเดียวกัน ลอง RUN โปรแกรม 21-7 ดูแล้วสังเกตตัวเลขที่ได้

```
10  REM  PRO#21-7
20  REM  RANDOM GROUP
30  FOR I = 2 TO 8 STEP 2
40  PRINT "RND(";I;")"
50  FOR J = 1 TO 6
60  PRINT  RND (I),
70  NEXT J: PRINT : NEXT I

]RUN
RND(2)
.935488986          .818647424
.0679459918         2.43769395E-04
.855436963          .101808286

RND(4)
.0845166788         .934070406
.735611717          .524775357
.422849342          .758975303

RND(6)
.338193568          .731304703
.154538185          .3636797
.741588325          .923538478

RND(8)
.204716841          .833726679
.773935711          .192536685
.92813501           .926721493
```

22 เทคนิคบางประการในการปรับปรุงโปรแกรม

22.1 ตัวอักษร ตัวเลข และเครื่องหมายทุกตัวในโปรแกรมนั้นมีความสำคัญสำหรับโปรแกรมมาก การพิมพ์ผิดพลาดหรือตกไป จะทำให้โปรแกรมนั้น ไม่ดำเนินไปตามจุดมุ่งหมาย ดังนั้น ควรจะพิมพ์ด้วยความระมัดระวัง ก่อนจะ RUN ควรจะ LIST มาตรวจสอบเสียก่อน การหยุดโปรแกรมเพื่อตรวจสอบนั้นทำได้โดยกด CTRL S เพื่อหยุด และ CTRL S อีกเพื่อเดินโปรแกรมต่อไป

22.2 ตัวอย่างโปรแกรมที่แสดงในหนังสือนี้ตั้งแต่ต้น ผู้เขียนมุ่งหวังจะแยกให้เห็นเด่นชัด จึงพยายามแยกเป็นบรรทัดย่อยๆ ซึ่งการกระทำดังนี้จะเป็นการง่ายที่จะแก้ หรือสอดบรรทัดเพิ่มเติมเข้าไป หากผู้อ่านมีความชำนาญพอแล้ว คำสั่งต่างๆ ที่แน่ใจว่าไม่มีการแก้แล้ว อาจจะนำมารวมกันอยู่ในบรรทัดเดียวกันหลายๆ คำสั่ง โดยมีเครื่องหมาย : แบ่งระหว่างคำสั่ง แต่ทั้งหมดจะต้องมีจำนวนคำไม่เกิน 239 ตัว มีข้อควรระวังว่าไม่มีคำสั่งอื่นตามหลังคำสั่ง IF... THEN..., GOTO หรือ REM เพราะโปรแกรมอาจจะข้ามคำสั่งเหล่านั้นไป โปรแกรมตัวอย่างต่อไปนี้เป็นโปรแกรมที่ผู้เขียนเขียนแยกให้เห็นชัด และโปรแกรมที่ย่อแล้วได้ผลเท่ากัน

โปรแกรมที่เขียนธรรมดา

```
5  REM  S16.1
10  HOME
20  INPUT "DO YOU WANT TO CONVERT
      'FOOT' TO'METER' ? (ANSWER
     YES OR NO)   ";A$
25  PRINT
30  IF A$ = "NO" THEN 110
35  IF A$ <  > "YES" THEN 20
40  PRINT
50  INPUT "O.K. THEN INPUT VALUE
     IN FT. ";F
60  GOSUB 1000
70  PRINT : PRINT
80  INPUT "ANY MORE ? (Y OR N) ";
     B$
85  PRINT
90  IF B$ = "Y" THEN 20
95  HOME
100  END
110  INPUT "O.K. THEN INPUT VALUE
      IN M.";M
120  GOSUB 2000
130  PRINT : PRINT
140  GOTO 80
150 :
1000  REM  CONVERT FT. TO M.
1010 X = F * .3048
1015  INVERSE : PRINT
1020  PRINT F;" FT. IS ";X;" M."
1025  NORMAL
1030  PRINT : PRINT
1040  RETURN
2000  REM  CONVERT M. TO FT.
2010 Y = M * 3.2808
2015  INVERSE : PRINT
2020  PRINT M;" M. IS ";Y;" FT."
2025  NORMAL
2030  PRINT : PRINT
2040  RETURN
```

โปรแกรมที่พยายามลดหมายเลขบรรทัดให้เหลือน้อยที่สุด

```
10  HOME
20  INPUT "DO YOU WANT TO CONVERT
     'FOOT' TO'METER' ? (ANSWER
    YES OR NO)   ";A$: PRINT : IF
    A$ = "NO" THEN 110
35  IF A$ <  > "YES" THEN 20
40  PRINT : INPUT "O.K. THEN INPU
    T VALUE IN FT. ";F: GOSUB 10
    00: PRINT : PRINT
80  INPUT "ANY MORE ? (Y OR N) ";
    B$: PRINT : IF B$ = "Y" THEN
    20
95  HOME : END
110  INPUT "O.K. THEN INPUT VALUE
     IN M.";M: GOSUB 2000: PRINT
    : PRINT : GOTO 80
1000  REM
1010 X = F * .3048: INVERSE : PRINT
    : PRINT F;" FT. IS ";X;" M."
    : NORMAL : PRINT : PRINT : RETURN

2000  REM
2010 Y = M * 3.2808: INVERSE : PRINT
    : PRINT M;" M. IS ";Y;" FT."
    : NORMAL : PRINT : PRINT : RETURN
```

22.3 การใช้ REM หรือหมายเหตุมากๆ มีทั้งดีและเสีย ที่ดีก็คือ เป็นการบอกขั้นตอนของโปรแกรม เพื่อให้ผู้อื่นได้รู้หรือตนเองเพื่อกันลืม แต่จะเป็นการเปลืองหน่วยความจำ วิธีที่เหมาะสมก็คือ ใช้ REM ไว้หลังคำสั่งอื่น เช่น 120 A = C+D : REM AREA จะเป็นการประหยัดหน่วยความจำได้

22.4 ตัวแปรที่ใช้ควรจะมีความหมาย ถึงแม้จะเขียนยาวไปบ้าง แต่ก็ช่วยไม่ให้เกิดการสับสน เช่น 150 INPUT SIDE แทนที่จะใช้ S เฉยๆ

22.5 ตัวแปรที่ใช้แล้วในตอนต้นของโปรแกรม อาจจะนำเอามาใช้อีกได้ถ้าแน่ใจว่าสิ่งที่บรรจุในตัวแปรนั้น ไม่ทำให้เกิดการผิดพลาดดังเช่น

```
30  FOR I = 1 TO 10
40
50
60
80  NEXT I
90
100
110  FOR I = 1 TO  LEN (A$)
120
130
180  NEXT I
```

22.6 หากมีตัวคงที่(CONSTANT) ซึ่งจะต้องนำมาใช้บ่อยครั้ง ให้บรรจุตัวคงที่นั้นไว้ในตัวแปร เครื่องจะทำงานเร็วขึ้นดังตัวอย่าง

```
10 PI = 3.14159
20  INPUT RADIUS
30 AREA = PI * RADIUS ^ 2
40 PERIMETER = 2 * PI * RADIUS
50  PRINT AREA,PERIMETER
```

22.7 การใช้ END ไว้บรรทัดหลังสุด มีความจำเป็นเพียงบางครั้งเท่านั้น โดยเฉพาะโปรแกรมที่มีโปรแกรมย่อย(SUBROUTINE) ต่อท้าย ปกติไม่ต้องใช้ END ก็ได้

22.8 ถ้าไม่มีการสับสนกัน เราอาจจะไม่ใช้ตัวแปรต่อท้ายคำว่า NEXT ก็ได้ (ใช้ NEXT เฉยๆ แทนที่จะเป็น NEXT I) เครื่องจะทำงานเร็วขึ้น

บทส่งท้าย

ผู้เขียนมีความมั่นใจว่า หากผู้อ่านได้อ่าน พร้อมกับได้ทดลองเขียนโปรแกรมและทำแบบฝึกหัดในหนังสือนี้มาตั้งแต่ต้นจนจบแล้ว ผู้อ่านสามารถจะนำความรู้มาใช้ในงานได้ แต่ผู้เขียนเสียใจที่จะเรียนว่า ความรู้เท่าที่มีอยู่เพียงที่ได้รับจากหนังสือนี้เป็นเพียงกระผีกหนึ่งของความรู้ที่ควรจะได้รับ ผู้อ่านควรจะศึกษาหาความรู้เพิ่มเติมเอง โดยเฉพาะจากหนังสือ 3 เล่ม ของผู้ผลิตเครื่องคอมพิวเตอร์ APPLE คือ "THE APPLESOFT TUTORIAL", "BASIC PROGRAMMING REFERENCE MANUAL" และ "THE DOS MANUAL" ซึ่งจะให้ความรู้ และความมั่นใจแก่ผู้อ่านเป็นอย่างมาก

ภาคผนวกที่ 1

คำที่จะนำมาใช้เป็นชื่อตัวแปรไม่ได้

&							
ABS	AND	ASC	AT	ATN			
CALL	CHR$	CLEAR	COLOR=	CONT	COS		
DATA	DEF	DEL	DIM	DRAW			
END	EXP						
FLASH	FN	FOR	FRE				
GET	GOSUB	GOTO	GR				
HCOLOR=	HGR	HGR2	HIMEM:	HLIN	HOME	HPLOT	HTAB
IF	IN#	INPUT	INT	INVERSE			
LEFT$	LEN	LET	LIST	LOAD	LOG	LOMEM:	
MID$							
NEW	NEXT	NORMAL	NOT	NOTRACE			
ON	ONERR	OR					
PDL	PEEK	PLOT	POKE	POP	POS	PRINT	PR#
READ	RECALL	REM	RESTORE	RESUME	RETURN	RIGHT$	
	RND	ROT=	RUN				
SAVE	SCALE=	SCRN(	SGN	SHLOAD	SIN	SPC(	
	SPEED=	SQR	STEP	STOP	STORE	STR$	
TAB(	TAN	TEXT	THEN	TO	TRACE		
USR							
VAL	VLIN	VTAB					
WAIT							
XPLOT	XDRAW						

ภาคผนวกที่ 2

การบันทึกโปรแกรมลงใน TAPE และ DISKETTE

1. ตามรูปที่ 1.1 หน่วยเก็บข้อมูล(STORAGE UNIT) สามารถจะเก็บข้อมูลที่เป็นโปรแกรมที่เราเขียนไว้ในหน่วยกรรมวิธี(PROCESSOR UNIT) ได้โดยการบันทึกไว้ในเทปแม่เหล็ก โดยใช้เครื่องบันทึกเทปธรรมดา หรือโดยการบันทึกไว้ในแผ่นคล้ายๆ กับจานเสียง โดยใช้เครื่องบันทึก(DISK DRIVE)

2. **การบันทึกโปรแกรมด้วยเทปแม่เหล็ก** เป็นวิธีที่ถูก เพราะเครื่องบันทึกหาได้ง่าย และเทปบันทึกก็ราคาถูก แต่การบันทึกและการนำเอามาใช้ค่อนข้างจะไม่สะดวก

การต่อสาย หลังเครื่อง APPLE จะมีรูเสียบอยู่สองรู มีอักษรบอก IN และ OUT ให้เสียบแจ๊คสาย IN ต่อเข้ากับรูเสียบสายหูฟังของเครื่องบันทึกเทป-และสาย OUT เข้ากับรูเสียบไมโครโฟนของเครื่องบันทึกเทป

การบันทึกโปรแกรมในเทป หมุนเทปในม้วนให้พร้อมที่จะบันทึกได้ พร้อมกับบันทึกหมายเลขจำนวนนับบนเทปไว้ เมื่อโปรแกรมในเครื่องคอมพิวเตอร์พร้อมที่จะบันทึกได้แล้ว ให้กดแป้นบนเครื่องคอมพิวเตอร์ SAVE อย่าเพิ่งกด RETURN เดินเครื่องเทปพร้อมกับกด RECORD เมื่อเทปเดิน แล้วจึงกดแป้น RETURN ถึงตอนนี้ เครื่องหมาย CURSOR บนจอภาพจะหายไป หลังจากนั้นประมาณ 10 ถึง 15 วินาที จะมีเสียงบี๊บดังขึ้น แสดงว่าการบันทึกกำลัง

ดำเนินไปอยู่ จนกระทั่งมีเสียงบี๊บอีกครั้ง แสดงว่าบันทึกหมดแล้วให้ม้วนเทป กลับมาจนถึงหมายเลขที่ตั้งต้น ตอนนี้ก็พร้อมที่จะนำข้อมูลมาใช้ได้แล้ว เทปยาวๆ สามารถจะบันทึกได้หลายโปรแกรม แต่วิธีที่ดีควรจะใช้เทปสั้นๆ บันทึกม้วนละโปรแกรม

การนำโปรแกรมในเทปมาใช้ เมื่อแน่ใจว่าเทปได้กลับมาที่จุดตั้งต้นแล้ว ให้กดแป้นว่า LOAD แล้วเดินเทป เมื่อเทปเริ่มเดินให้กด RETURN CURSOR บนจอจะหายไป หลังจาก 5 ถึง 20 วินาทีผ่านไปจะมีเสียงบี๊บดังขึ้น แสดงว่าเทปเริ่มถ่ายทอดโปรแกรมให้แก่เครื่อง เมื่อหมดโปรแกรมจะมีเสียงบี๊บดังอีกครั้งหนึ่ง พร้อมกับเครื่องหมาย CURSOR กระพริบขึ้นมา เมื่อจะใช้โปรแกรมนั้นก็กด RUN โปรแกรมจะดำเนินไปตามที่ได้วางไว้ หากการกระทำตั้งแต่ต้นไม่ได้ผล ลองปรับความดังของเสียงที่เครื่องบันทึกเทป แล้วทดลองใหม่ตั้งแต่ต้น

3. **การบันทึกโปรแกรมด้วยแผ่นจาน** การใช้แผ่นจาน (DISKETTE) ใช้ได้สะดวก และรวดเร็วกว่าใช้เทปบันทึกเสียงมาก แต่เครื่องบันทึกพร้อมกับแผ่นจานมีราคาแพง และต้องใช้อย่างระมัดระวัง อย่าทำเครื่องบันทึกตก หรือทำของอื่นตกทับ และควรจะตั้งให้ห่างจากเครื่องรับโทรทัศน์อย่างน้อย 2 ฟุต มิฉะนั้นคลื่นแม่เหล็กจากเครื่องรับโทรทัศน์ อาจจะทำให้เครื่องเสียได้ ตัวแผ่นจานเป็นแผ่นพลาสติก มีเส้นผ่าศูนย์กลางประมาณ 5 นิ้ว ผิวเคลือบด้วยสารแม่เหล็ก แผ่นจะบรรจุอยู่ในซองพลาสติก อย่าพยายามเปิดซองนั้น อย่าดัดแผ่นงอ อย่าจับส่วนของจานที่แสดงออกมาตรงรูที่เจาะซอง อย่าตากแดด หรือถูกอบด้วยความร้อน เช่น เก็บไว้ในรถที่จอดตากแดด เป็นต้น อย่าให้ถูกฝุ่น อย่าให้ถูกน้ำ อย่าให้เข้าใกล้แม่เหล็ก และ

อย่า ฯลฯ อีกหลายอย่างที่จะทำให้แผ่นจานเสีย เวลาเขียนชื่อลงบนซอง อย่าใช้ปากกาลูกลื่น ให้ใช้ปากกาเมจิคที่มีปลายอ่อน

การต่อเครื่องบันทึกกับเครื่องคอมพิวเตอร์ เครื่องบันทึกจะต่อเข้ากับวงจรที่เรียกว่า CONTROLLER CARD และตัว CARD นี้จะเสียบอยู่ในช่อง (SLOT) ซึ่งปกติจะอยู่ที่ช่องหมายเลข 6 CARD หนึ่งใบจะบรรจุเครื่องบันทึกได้ 2 เครื่อง ซึ่งจะมีหมายเลขเรียกว่า DRIVE 1 และ DRIVE 2

การบันทึกโปรแกรมลงบนแผ่นจาน ถ้าแผ่นจาน (DISKETTE) เป็นแผ่นจานใหม่ เราจะต้องทำการเตรียมที่เรียกว่า INITIALIZE แผ่นจานก่อนโดยการกระทำดังนี้

1. สอดแผ่นจานเข้าในเครื่อง โดยการจับที่มุมขวาของแผ่นจานที่มีกระดาษขาวสำหรับเขียนชื่อ จับเบาๆ สอดไปให้สุดแล้วปิดฝา
2. เปิดสวิทช์คอมพิวเตอร์พร้อมกับเปิดสวิทช์จอภาพ เครื่องบันทึกจะหมุนพร้อมกับไฟแดงจะติดและจะหมุนอยู่เรื่อยไปไม่มีวันหยุด ให้กดปุ่ม RESET เครื่องบันทึกจะหยุดพร้อมกับ CURSOR จะปรากฏบนจอภาพ เมื่อมาถึงตอนนี้เราสามารถจะเขียนโปรแกรมได้ แต่ไม่สามารถจะบันทึกในแผ่นจานใหม่ได้ ให้ทำต่อไปดังนี้
3. เปิดฝาเครื่องบันทึกออก ถอดแผ่นจานใหม่ออก ใส่แผ่นจานที่มีชื่อ "SYSTEM MASTER" (ที่จะแถมมาพร้อมกับเครื่อง) ปิดฝา, พิมพ์คำว่า PR#6 แล้วกด RETURN คราวนี้ไฟแดงที่เครื่องบันทึกจะติดสักครู่ก็จะดับ
4. พิมพ์คำว่า NEW แล้วพิมพ์โปรแกรมง่ายๆ ลงไป เช่น

```
10  PRINT "DISKETTE INIT. BY...."
20  PRINT "ON JAN. 17, 1983"
30  END
```

ช่องว่างนั้น พิมพ์ชื่อของท่านลงไป แล้วลอง RUN โปรแกรมดูจนเป็นที่พอใจ ถอด "SYSTEM MASTER" ออก ใส่แผ่นจานใหม่ไปแทนที่

5. กด INIT HELLO [RETURN] คำว่า HELLO จะใส่อะไรลงไปแทนก็ได้ ถึงตอนนี้เครื่องบันทึกจะหมุนอยู่สักพัก แล้วหยุด เป็นอันว่าหมดพิธี แล้วเราสามารถจะใช้แผ่นจานนี้บันทึกโปรแกรมที่เราเขียนไปได้

ลองหัดบันทึกโปรแกรมดังนี้(ตอนนี้สมมติว่าเครื่องปิดอยู่) เปิดฝาเครื่องบันทึก สอดแผ่นเข้าไป แล้วปิดฝา ต่อไปเปิดสวิทช์เครื่องคอมพิวเตอร์ เครื่องบันทึกจะหมุนพร้อมกับไฟแดงติดสักครู่แล้วจึงดับ บนจอภาพจะปรากฏภาพดังนี้

```
DISKETTE INIT BY PRAPAT
ON JAN.17,1983
```

CURSOR จะปรากฏที่ส่วนล่างซ้ายของจอภาพ กด NEW เพื่อลบโปรแกรมที่มีอยู่ออกแล้วลองพิมพ์โปรแกรมต่อไปนี้

```
10  FOR I = 1  TO  10
20  PRINT I
30  NEXT I
```

เมื่อกด RUN ตัวเลข 1 ถึง 10 จะปรากฏบนจอเป็นแถวเรียงลงไปข้างล่าง ต่อไปกด SAVE COUNT เครื่องบันทึกจะหมุนพร้อมกับไฟแดงติดสักครู่ จึงหยุดหมุนพร้อมกับไฟดับ คำว่า COUNT คือชื่อของโปรแกรมซึ่งจำเป็นสำหรับการบันทึกด้วยแผ่นจาน

ลองเอาโปรแกรมในแผ่นจานมาใช้ ด้วยการปิดสวิทช์เครื่องก่อนโปร-แกรมที่มีอยู่ในเครื่องจะหายไป เปิดเครื่องใหม่ เครื่องบันทึกจะหมุนสัก

ครูก็หยุด บนจอภาพจะปรากฏดังนี้

```
DISKETTE INIT BY PRAPAT
ON JAN.17,1983
```

ต่อไปพิมพ์คำว่า CATALOG [RETURN] บนจอภาพจะแสดงบรรทัดต่อไปดังนี้

```
DISK VOLUME 254
A 002 HELLO
A 002 COUNT
```

กด LOAD COUNT เครื่องบันทึกจะหมุน ตอนนี้โปรแกรมจะถ่ายจากแผ่นจานลงในเครื่อง กด LIST จะได้โปรแกรม COUNT ออกมา ต่อไปกด RUN เครื่องจะแสดงเลข 1 ถึง 10 ออกมา แสดงว่าโปรแกรมใช้ได้ หรือเราอาจจะกด RUN COUNT ก็ได้ เครื่องจะ LOAD โปรแกรมจากแผ่นจานลงเครื่องแล้ว RUN ให้เลย

หนังสืออ้างอิง

1. APPLESOFT TUTORIAL.-APPLE COMPUTER INC.
2. BASIC PROGRAMMING REFERENCE MANUAL.-APPLE COMPUTER INC.
3. THE DOS MANUAL.-APPLE COMPUTER INC.
4. APPLESOFT LANGUAGE.-BRIAN D.BLACKWOOD & GEORGE H. BLACK-WOOD
5. APPLE II USER'S GUIDE.-LON POOLE WITH MARTIN MC. NIFF & STEVEN COOK
6. TALKING TO A COMPUTER IN BASIC.-WONDER T. LEONG
7. COMPUTING USING BASIC.-TONIA COPE
8. BASIC PROGRAMMING SELF-TAUGHT.-SEYMOUR C. HIRSCH
9. PROGRAMMING WITH BASIC. -BYRON S. GOTTFRIED
10. I SPEAK BASIC TO MY APPLE.-AUBREY B. JONES, JR.
11. KIDS & THE APPLE.-EDWARD H. CARLSON.

ดรรชนี

หนังสือสำหรับผู้ศึกษา/ใช้/เล่นคอมพิวเตอร์



โปรแกรมคอมพิวเตอร์ ภาษาเบสิค

โดย ยืนภู่วรวรรณ พิชิต สุขเจริญพงศ์ สมนึก คีรีโต และ ภิญโญ ศรีสุขินวงศ์ ราคา 55 บาท

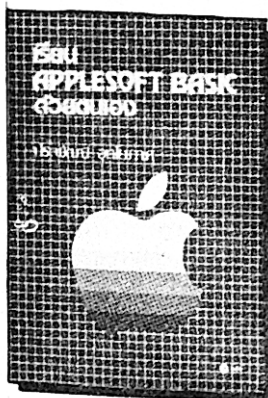
เป็นหนังสือภาษาเบสิคอีกเล่มหนึ่งที่จัดทำขึ้น โดยคณะอาจารย์จากห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตร โดยพยายามสร้างพื้นฐานการเขียนโปรแกรมสำหรับผู้เริ่มต้น การประยุกต์ใช้งานของภาษาเบสิคในเรื่องการเขียนภาพ การสร้างเกม การใช้งานทางคณิตศาสตร์และสถิติ การใช้งานทางด้านธุรกิจการบริหาร



คอมพิวเตอร์กับการเขียนโปรแกรมภาษาเบสิค

โดย พรทิพย์ เลาหวิโรจน์ และสุพจน์ จิตต์ประเสริฐ ราคา 65 บาท

หนังสือเล่มนี้สำหรับผู้ที่ต้องการเรียนและเขียนภาษา APPLESOFT BASIC ด้วยตนเองให้เป็นภายใน 50ชั่วโมงโดยจะเริ่มต้นตั้งแต่พื้นฐานของการโปรแกรมด้วยภาษาเบสิค อ้างอิงถึงการใช้งานกับเครื่องยี่ห้อ APPLE โดยจะค่อย ๆ พัฒนาคุณขึ้นมาจนถึงระดับที่จะโปรแกรมงานบางอย่างได้เอง



เรียน APPLESOFT BASIC ด้วยตนเอง (เล่ม 1)

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 45 บาท

หนังสือเล่มนี้จะช่วยให้ท่านสามารถเขียนโปรแกรมสั่งงานไมโครคอมพิวเตอร์ได้ด้วยตัวท่านเอง โดยจะเริ่มตั้งแต่คอมพิวเตอร์คืออะไร มีอะไรบ้าง จะต้องสั่งงานอย่างไร คำสั่งที่ใช้มีอะไรบ้าง ตลอดเล่มจะมีโปรแกรมที่ทำการทดสอบแล้วโดยผ่านเครื่อง APPLE II ให้ท่านฝึกฝนโดยเริ่มจากง่าย ๆ ขึ้นไป จนในที่สุดท่านก็จะสามารถอ่านโปรแกรมเข้าใจได้ด้วยตัวท่านเอง



เรียน APPLESOFT BASIC ด้วยตนเอง (เล่ม 2)

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 42 บาท

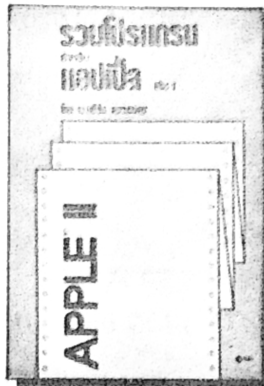
เป็นหนังสือเพิ่มความรู้สำหรับผู้ที่อ่านเล่ม 1 มาแล้ว เพื่อจะได้ศึกษาส่วนที่ผู้เขียนเว้นไว้เพื่อมิให้สับสนในระยะที่เริ่มเรียนรู้ บทต่าง ๆ ภายในเล่มมี ลึกเข้าไปในคอมพิวเตอร์ ; เพิ่มเติมเสริมแต่ง, FUNCTIONS, กราฟิก, SHAPE TABLE, TEXT FILE และ FLOW CHART



เรียน / เล่น / ใช้ โปรแกรม APPLESOFT BASIC

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 45 บาท

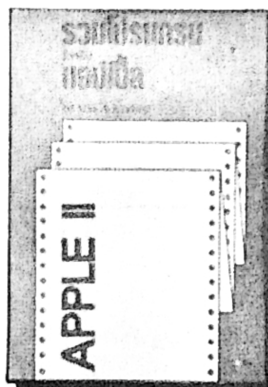
เมื่อท่านสามารถอ่านโปรมแกรมเข้าใจแล้ว หนังสือเล่มนี้จะเป็นเล่มต่อไปที่ช่วยเพิ่มประสบการณ์แก่ท่านในด้านการหัดป้อนข้อมูล (KEY) หัด SAVE, LOAD, LIST และ RUN หัดแก้และหัดแปลงโปรแกรมซึ่งต่อไปก็จะสามารถเขียนโปรแกรมได้เอง สำหรับโปรแกรม 40 โปรแกรมที่ได้รวบรวมไว้ในเล่มมีทั้งที่เป็นเกม โปรแกรมทางการศึกษา คณิตศาสตร์ สถิติ และธุรกิจ เพราะฉะนั้นท่านสามารถจะเลือกฝึกได้อย่างจุใจ



รวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 1

โดย เฮาเอิร์ด เบเรนบอน ราคา 48 บาท

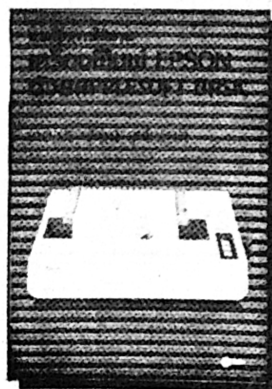
และนี่คือ โปรแกรมใช้งานสำหรับนักเล่นไมโครคอมพิวเตอร์ที่ต้องการจะใช้เครื่อง APPPLE II ให้คุ้มค่า มี 28 โปรแกรมอยู่ภายในเล่มแบ่งเป็น 6 ภาคตามลักษณะการใช้งาน เช่น ภาคเกี่ยวกับเวลา ธุรกิจและการลงทุน การศึกษา เป็นต้น ตัวอย่างโปรแกรม ได้แก่เครื่องควบคุมการปิดเปิดไฟฟ้าสารพัดประโยชน์ เครื่องตั้งเวลาอัตโนมัต และคิดค่าเสื่อมราคา วิเคราะห์การใช้ไฟฟ้า หมอดูไพ่ป๊อก และเกมฝึกสมองอีกมากมาย



รวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 2

โดย เฮาเอิร์ด เบเรนบอน ราคา 55 บาท

เป็นหนังสือรวมโปรแกรมใช้งานเครื่อง APPLE II อีก 37 โปรแกรมอาทิ เช่น การคิดดอกเบี้ยทบต้น โปรแกรมเก็บหมายเลขโทรศัพท์ โปรแกรมรักษาสภาพสมดุลของบัญชีเช็ค โปรแกรมแนะนำการซื้อหุ้น ทดสอบความสามารถทางจิต โปรแกรมแบ่งเกรดนักเรียนและโปรแกรมทดสอบความสามารถอีกมากมาย



เทคนิคการใช้งานเครื่องพิมพ์ EPSON ด้วย APPLESOFT BASIC

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 22 บาท

หนังสือเล่มนี้จะบอกวิธีใช้เครื่องพิมพ์ EPSON ที่พ่วงกับเครื่องคอมพิวเตอร์ตระกูล APPLE มีการสร้างตัวหนังสือแบบต่าง ๆ ถึง 11 แบบ การปรับระยะบรรทัด การพิมพ์ภาพกราฟิกพร้อมกับมีโปรแกรมแถมให้ การพิมพ์นามบัตรสำเร็จรูป การพิมพ์กราฟแท่ง มีตารางสรุปคำสั่งต่าง ๆ อย่างที่จะหาไม่ได้ในหนังสือทั้งไทยและเทศ และท้ายเล่มมีโปรแกรมสั้น ๆ สนุก ๆ สำหรับเด็ก ๆ หนังสือเล่มนี้เหมาะสำหรับผู้ที่มีเครื่องพิมพ์ EPSON อยู่แล้วหรือผู้ที่กำลังคิดจะมี



ระบบจัดการฐานข้อมูล dBASE II

โดย ยืน ภู่วรวรรณ และสมนึก คีรีโต ราคา 60 บาท

เป็นหนังสือสำหรับผู้ที่ยังไม่เคยใช้ไมโครคอมพิวเตอร์ในระบบจัดการฐานข้อมูลมาก่อนเลย และสนใจการใช้งาน dBASE II ซึ่งเป็นโปรแกรมที่มีประสิทธิภาพสูงสุดและได้รับความนิยมมาก ภายในเล่มแบ่งเป็น 3 ตอน ตอนแรกจะวางพื้นฐานของระบบจัดการฐานข้อมูล ตลอดจนโปรแกรมโครงสร้าง ตอนที่สองว่าด้วยวิธีการทำงานของ dBASE II และตอนที่สามว่าด้วยการนำเอา dBASE II มาใช้งาน



อิเล็กทรอนิกส์เวอร์คชีต

โดย ยืน ภู่วรวรรณ และอัศนีย์ ก่อตระกูล ราคา 59 บาท

เหมาะสำหรับผู้ที่เรียนรู้การใช้งานคอมพิวเตอร์มาบ้างแล้ว และกำลังต้องการนำไมโครคอมพิวเตอร์มาใช้งานจริง โดยจะให้หลักการและวิธีใช้อิเล็กทรอนิกส์เวอร์คชีตซึ่งใช้ได้ทั้งวิสิแคลค์ มัลติแพลน ซูเปอร์แคลค์ (ทั้งกำลัง 2 และกำลัง 3) และโลตัส 123 ตัวอย่างของอิเล็กทรอนิกส์เวอร์คชีตในหนังสือเล่มนี้ใช้กับโปรแกรมซูเปอร์แคลค์เป็นหลัก แต่ก็มีหลักการใช้หรือการเรียกคล้ายอิเล็กทรอนิกส์เวอร์คชีตอีกหลายตัว



เรียนมัลติแพลนด้วยตนเอง

โดย ประพัฒน์ อุทโยภาศ ราคา 46 บาท

หากท่านมีนิ้วมือเหลืออย่างน้อยสองนิ้ว พออ่านภาษาอังกฤษออก มีไมโครคอมพิวเตอร์ที่มีหน่วยความจำอย่างน้อย 64 K BYTE มีโปรแกรมสำเร็จรูปชื่อมัลติแพลน และได้อ่านหนังสือเล่มนี้เพียงครึ่งเล่ม ท่านก็สามารถนำคอมพิวเตอร์ไปใช้ประโยชน์ได้อย่ากว้างขวาง อาทิเช่น ทำบัญชีสินค้าคงคลัง ออกอินวอยซ์ เก็บหมายเลขโทรศัพท์ ฯลฯ หนังสือเล่มนี้ใช้ภาษาง่าย ๆ อย่างเป็นกันเองเหมือนเพื่อนคุยกัน แต่ขณะเดียวกันจะให้ความรู้ที่แม้แต่ในหนังสือคู่มือการใช้โปรแกรมนี้เองก็ไม่ได้กล่าวไว้ กล่าวได้ว่าหนังสือเล่มนี้จะเป็นประโยชน์แก่ทุกคนตั้งแต่ผู้เริ่มใช้ จนกระทั่งผู้ที่ได้เคยใช้โปรแกรมนี้มาอย่างจัดเจน



โปรแกรมคอมพิวเตอร์ภาษาฟอร์แทรน

โดย ยืน ภู่วรวรรณ พิชิต สุขเจริญวงศ์ และนนทวัฒน์ จันทร์เจริญ ราคา 68 บาท

หนังสือเล่มนี้เหมาะสำหรับใช้เป็นตำราในระดับวิทยาลัย มหาวิทยาลัย ตลอดจนผู้ที่จะศึกษาด้วยตนเอง เนื้อหาภายในได้วางโครงร่างในลักษณะต่อ

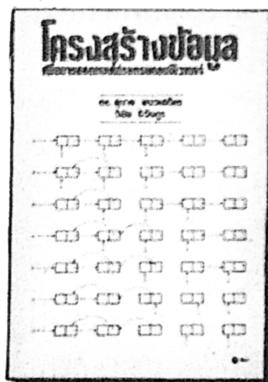
เรียงกันไปโดยเริ่มจากหลักการทางคอมพิวเตอร์ สิ่งที่ควรรู้เกี่ยวกับคอมพิวเตอร์ หลักการทางด้านภาษาคอมพิวเตอร์ การเขียนแผนภูมิสายงานหรือผังงานการใช้คำสั่งต่าง ๆ ที่มีในภาษาฟอร์แทรนและตอนสุดท้ายว่าด้วยการประยุกต์ภาษาฟอร์แทรนมาใช้งาน



โปรแกรมคอมพิวเตอร์เบื้องต้น ภาษาโฟแทรน 4

โดย เดือน สินธุพันธ์ประทุม ราคา 49 บาท

เป็นหนังสืออีกเล่มหนึ่งที่ได้รับความสนใจอย่างสูงจากผู้อ่านจนต้องพิมพ์เป็นครั้งที่ 4 เนื่องจากเนื้อหาสามารถอ่านเข้าใจได้ง่ายทั้งผู้เริ่มเรียนหรือผู้ที่พอมีพื้นความรู้มาบ้างแล้วคำสั่งพื้นฐานจากหนังสือเล่มนี้สามารถใช้ได้กับคอมพิวเตอร์ทุกเครื่องที่รับภาษาโฟแทรน 4 ได้ และยังเพิ่มเติมคำสั่งบางประเภทซึ่งใช้ได้เฉพาะเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงด้วยในบทที่ 10



โครงสร้างข้อมูล

โดย สุขาย ธนวเสถียร และ วิชัย จิวังกูร ราคา 54 บาท

หลังจากที่เรียนภาษาคอมพิวเตอร์ตลอดจนการเขียนโปรแกรมแล้ว ขั้นต่อไปคือเรียนวิชาโครงสร้างข้อมูล วิชานี้มีความสำคัญอย่างมากเนื่องจากเป็นวิชาการพื้นฐานทางคอมพิวเตอร์ที่จะช่วยให้ผู้เขียนโปรแกรมทุกคนสามารถใช้โครงสร้างข้อมูลแบบที่เหมาะสม ซึ่งช่วยให้โปรแกรมทำงานเร็วขึ้นกินเนื้อที่น้อยลง และมีประสิทธิภาพตามที่ต้องการ ฉะนั้นถ้าท่านได้เริ่มเรียนรู้การใช้คอมพิวเตอร์โดยใช้เครื่องไมโครไม่ว่าจะเป็น APPLE จริงหรือเทียมหรือเครื่องยี่ห้อใดก็ตาม หนังสือเล่มนี้จะเป็นเล่มที่คุณพลาดไม่ได้



ไมโครโปรเซสเซอร์ - ไมโครคอมพิวเตอร์

โดย ยืน ภู่วรวรรณ และ วัฒนา เชียงกูล 75 บาท

หนังสือเล่มนี้เป็นการวางพื้นฐานของไมโครคอมพิวเตอร์ โดยเน้นให้เห็นถึงโครงสร้างคอมพิวเตอร์ที่จำเป็นต้องมีซีพียูหน่วยความจำ และอินพุท-เอาท์พุท นอกจากนี้ยังกล่าวถึงรายละเอียดของซีพียู Z-80 และการต่อเป็นระบบไมโครคอมพิวเตอร์ การเขียนโปรแกรม การประยุกต์ใช้งาน และสุดท้ายจะเป็นบทที่ว่าด้วยอุปกรณ์สนับสนุน ระบบไมโครคอมพิวเตอร์ที่สำคัญซึ่งได้แก่ 8214, 8212, Z-80 PIO, Z-80 CTC และ 8251



เรื่องน่ารู้เกี่ยวกับไมโครคอมพิวเตอร์

โดย ยืน ภู่วรวรรณ ราคา 53 บาท

เหมาะสำหรับผู้ที่อยากรู้รายละเอียดทั่วไปเกี่ยวกับไมโครคอมพิวเตอร์ เพื่อศึกษาข้อมูลรายละเอียดต่าง ๆ โดยเนื้อหาจะเน้นในลักษณะเรื่องทั่วไปที่เกี่ยวข้องถึงฮาร์ดแวร์ ซึ่งได้แก่ส่วนประกอบต่างๆ และส่วนของซอฟท์แวร์ที่ใช้กับไมโครคอมพิวเตอร์ จึงเปรียบเสมือนคู่มือที่ผู้สนใจในระดับพื้นฐานควรมีไว้เพราะสามารถศึกษารายละเอียดต่าง ๆ รวมทั้งเป็นข้อมูลสำหรับการจัดซื้อ การพิจารณาสเปค และสร้างฐานความรู้เพื่อศึกษาไมโครคอมพิวเตอร์ในระดับที่ลึกซึ้งขึ้นไป

• สมาชิกวารสารของซีเอ็ดทุกฉบับสั่งซื้อได้ในราคา 15% ค่าจัดส่งฟรี !!

หาซื้อและเลือกชมได้ที่ ร้านหนังสือย่านสยามสแควร์ วังบูรพา สนามหลวง ศูนย์หนังสือจุฬาฯ ร้านเมฆรุ่งเรือง ร้านเพ็ญบุญ ร้านดวงกมลสมัย

หรือสั่งซื้อทางไปรษณีย์ โดยส่งธนาณัติสั่งจ่าย ปท.ดินแดง ในนามบริษัทซีเอ็ดยูเคชั่น จำกัด 800/43-45 ซอยตระกูลสุข ถนนอโศกดินแดง เขตห้วยขวาง



เรียน APPLESOFT BASIC ด้วยตนเอง

หนังสือเล่มนี้ มีวัตถุประสงค์ที่จะให้ความรู้แก่ผู้ที่ไม่มีพื้นฐานทางคอมพิวเตอร์มาก่อน ให้สามารถเขียนโปรแกรมได้ สามารถใช้คอมพิวเตอร์เพื่อผลประโยชน์ในการทำงานและการให้ความบันเทิง แต่มิได้มีจุดมุ่งหมายที่จะทำให้ผู้อ่านสามารถประกอบอาชีพในการเขียน-โปรแกรม หรือมิได้มีจุดมุ่งหมายที่จะใช้เป็นเอกสารอ้างอิงสำหรับผู้ที่ได้เล่าเรียนมาทางด้านคอมพิวเตอร์โดยตรง ผู้ที่จะได้รับประโยชน์จากหนังสือเล่มนี้ คือ ผู้ที่อยากจะเรียนรู้เรื่องคอมพิวเตอร์ แต่ไม่มีเวลาที่จะเข้าเรียน สามารถจะเรียนด้วยตนเอง จนมาใช้ให้เกิดประโยชน์ได้

ภาษา BASIC ในหนังสือนี้ทั้งหมด สามารถนำมาใช้กับเครื่อง Apple II ได้ทันที และโปร-แกรมทั้งหมดที่เขียนไว้ในหนังสือนี้ เขียนด้วยเครื่อง Apple II ผ่านทางเครื่อง Epson Printer โดยได้ทำการทดสอบจนเป็นที่แน่ใจแล้วว่าสามารถให้ผลได้จริง ๆ



พ.ต.ประพัฒน์ อุทโยภาศ จบปริญญาตรี และโท ทางวิศวกรรมศาสตร์ จากจุฬาลงกรณ์มหาวิทยาลัย สมัยที่ ณ ณรยังไม่มีการันต์ ในสาขาวิชาที่ไม่เกี่ยวแก่คอมพิวเตอร์ เพราะสมัยนั้นยังไม่มีการสอนกัน จบออกมาก็ทำงานอยู่กรมโยธา ซึ่งสมัยนั้นมีห้อยท้ายว่า "เทศบาล" แทนที่จะเป็น "ธิการ" ทำให้สับสนกับเทศบาลนครกรุงเทพบ่อย ๆ อยู่มาได้ 3-4 ปี ขอโอนไปสอนหนังสืออยู่โรงเรียนนายร้อยพระจุลจอมเกล้า อยู่ได้ถึง 10 ปี จนได้รับพระราชทานยศเป็นพันตรี บริษัทฯฝรั่งเสนอจ้างในเงินเดือนที่สูงกว่าจอมพล จึงทนเย้ายวนใจไม่ไหว อยู่ได้ 9 ปีกว่า ๆ เงินเดือนชักเยอะ แต่เขี้ยวชักจะคลอน เลยต้องออกมา เดี๋ยวนี้เป็นรองผู้จัดการ บริษัท กรุงเทพเอ็นยิเนียริ่งคอนซัลแตนท์ จำกัด ตั้งใจว่าจะอยู่ไปจนเกษียณ ถ้าไม่ถูกเขาไล่ออกเพราะหมกมุ่นกับ Apple มากเกินไป

ISBN 974-7964-77-5